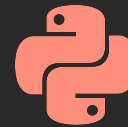




Dicas para criar um projeto Python

Live de Python # 192



1. Estrutura de diretórios

E mais algumas coisas importantes

2. Ferramentas de desenvolvimento

Facilitando a vida

3. Ferramentas de checagem

Padronizando código entre times

4. Automações

Algumas maneiras facilitar a vida

Como Começar um Projeto Python Perfeito

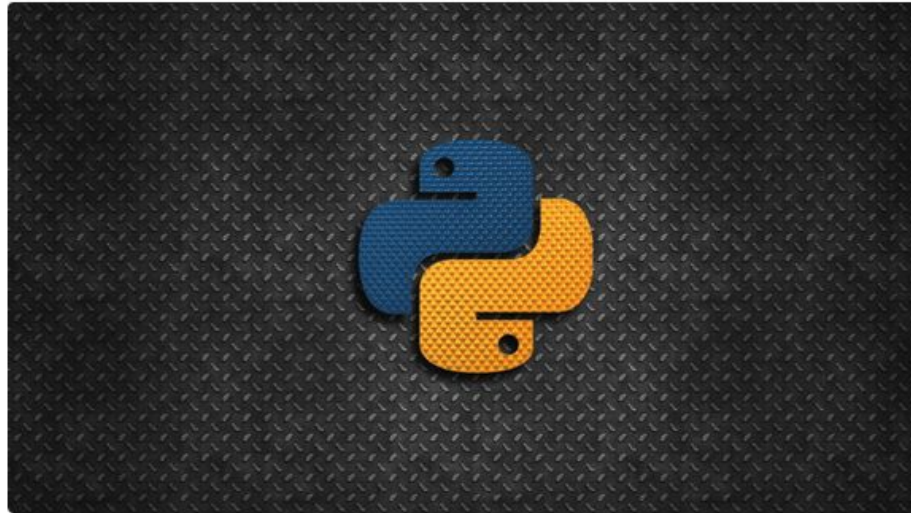
André Felipe Dias

2022-01-05

[1 Comment](#)

Também disponível em:

[English](#)



Aviso!



<https://blog.pronus.io/posts/python/como-comecar-um-projeto-python-perfeito/>

Projetos de natureza exploratória*



Aviso 2!





picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Acássio Anjos, Ademar Peixoto, A Earth, Alexandre Harano, Alexandre Santos, Alexandre Takahashi, Alexandre Tsuno, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Amaziles Carvalho, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rocha, Arnaldo Turque, Artur Zalewska, Bruno Barcellos, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, César Almeida, Clara Battesini, Cleber Santos, Dartz Dartz, David Kwast, Dhyeives Rodovalho, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Donivaldo Sarzi, Douglas Bastos, Douglas Braga, Douglas Martins, Douglas Zickuhr, Eli Júnior, Emerson Rafael, Eric Niens, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Fabiano Gomes, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Flávio Meira, Flavkaze Flavkaze, Francisco Alencar, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Pereira, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Hugo Bacilli, Igor Taconi, Ismael Ventura, Israel Gomes, Italo Silva, Jair Andrade, Janael Pinheiro, João Lugão, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, Jorge Plautz, José Gomes, Jose Mazolini, José Prado, Juan Gutierrez, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Miranda, Leonardo Cruz, Leonardo Mello, Leonardo Nazareth, Lidianne Monteiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Luciano Silva, Luciano Teixeira, Luiz Junior, Maiquel Leonel, Marcela Campos, Marcelino Pinheiro, Márcio Martignoni, Marco Mello, Marcos Ferreira, Marcos Mendonça, Marco Yamada, Maria Clara, Marina Passos, Mario Deus, Matheus Vian, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Nilton Pimentel, Osvaldo Neto, Patricia Minamizawa, Patrick Brito, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, Priscila Santos, Rafael Barbosa, Rafael Lopes, Reinaldo Silva, Renan Moura, Renato Veirich, Richard Nixon, Riverfount Riverfount, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Junior, Rodrigo Vaccari, Rogério Sousa, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sara Selis, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Tony Dias, Tyrone Damasceno, Victor Wildner, Vinícius Bastos, Vinicius Figueiredo, Vítor Gomes, Vlademir Souza, Vladimir Lemos, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Duarte, Yuri Fialho, Yury Barros



Obrigado você



A base do projeto

Estrut
ura

No que se baseia uma boa estrutura?



- Separação de responsabilidades
 - Cada coisa deve estar no seu lugar

Nosso projeto

No que se baseia uma boa estrutura?



- Separação de responsabilidades
 - Cada coisa deve estar no seu lugar

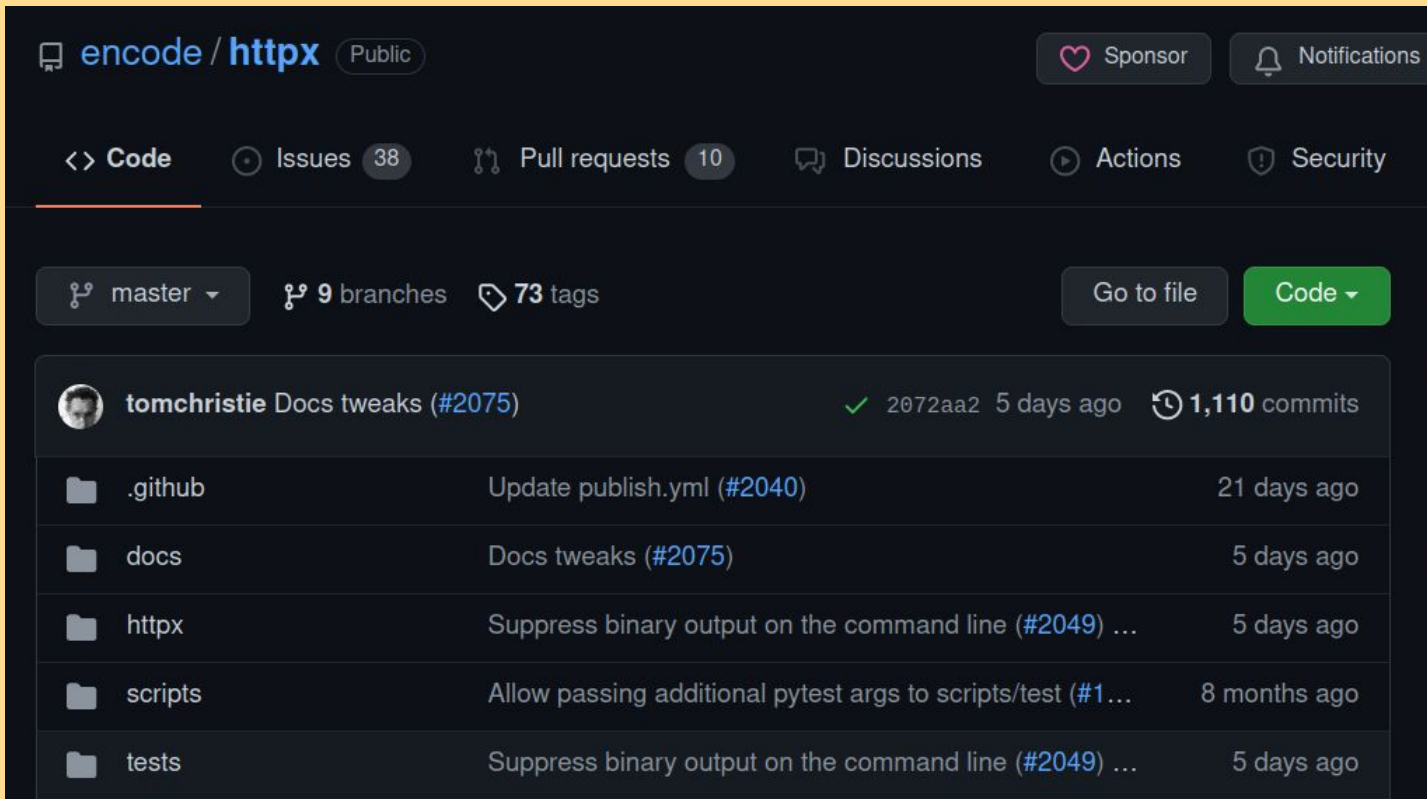
Documentação

Código Fonte

Testes

Nosso projeto

Olhando a organização de projetos



The screenshot shows the GitHub repository page for `encode/httpx`. The repository is public and has 38 issues, 10 pull requests, and 1,110 commits. The main branch is `master`, and there are 9 other branches and 73 tags. The repository is organized into several folders: `.github`, `docs`, `httpx`, `scripts`, and `tests`. The `docs` folder is highlighted, showing a commit by `tomchristie` titled "Docs tweaks (#2075)" made 5 days ago.

encode / httpx Public

Sponsor Notifications

<> Code Issues 38 Pull requests 10 Discussions Actions Security

master 9 branches 73 tags Go to file Code

tomchristie Docs tweaks (#2075) 2072aa2 5 days ago 1,110 commits

.github	Update publish.yml (#2040)	21 days ago
docs	Docs tweaks (#2075)	5 days ago
httpx	Suppress binary output on the command line (#2049) ...	5 days ago
scripts	Allow passing additional pytest args to scripts/test (#1...	8 months ago
tests	Suppress binary output on the command line (#2049) ...	5 days ago

<https://github.com/encode/httpx>

Olhando a organização de projetos



tiangolo / **fastapi** Public Sponsor Notifications

[Code](#) [Issues 922](#) [Pull requests 468](#) [Discussions](#) [Actions](#) [Projects](#)

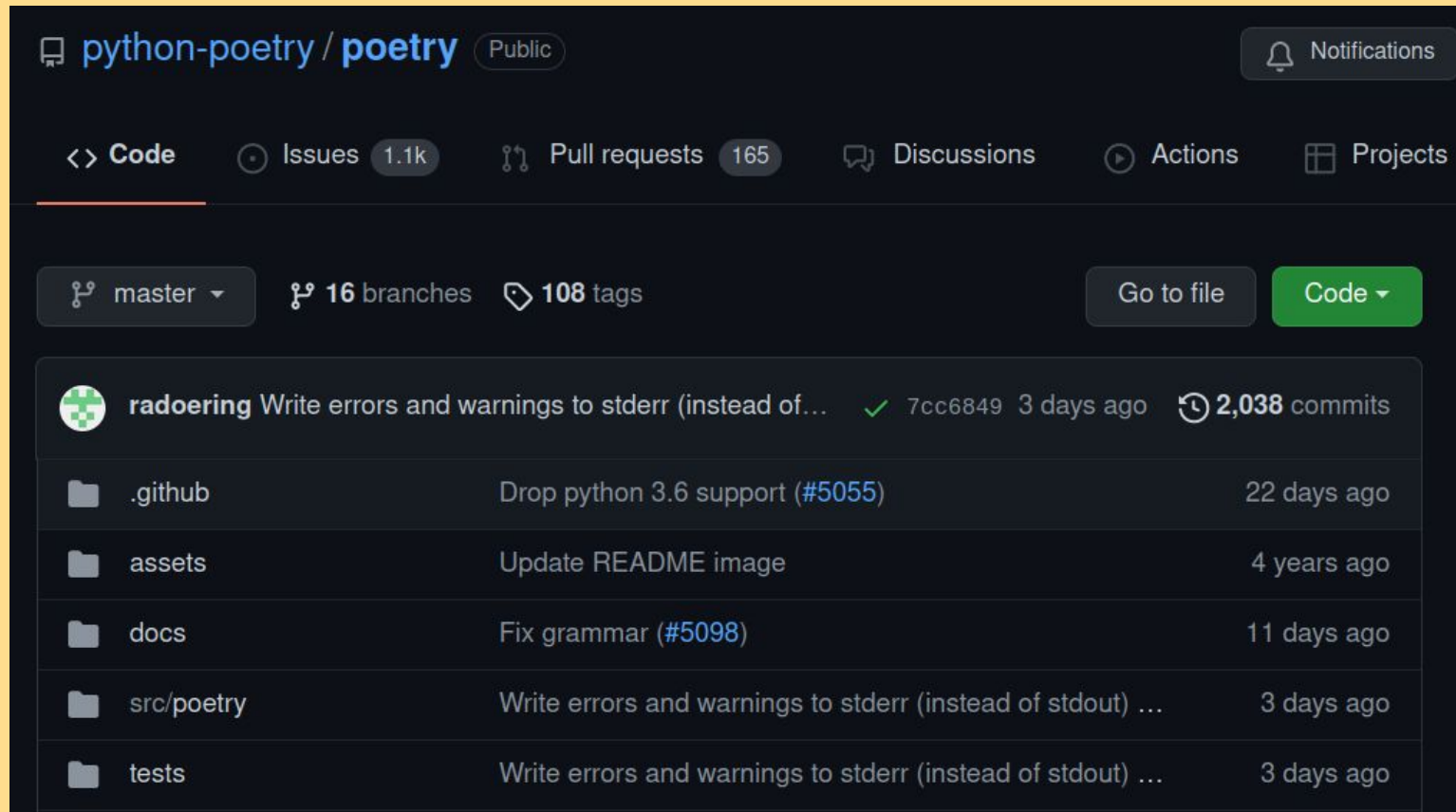
[master](#) [1 branch](#) [120 tags](#) [Go to file](#) [Code](#)

github-actions Update release notes ✓ 78b07cb 20 hours ago 🕒 1,961 commits

.github	Only build docs on push when on master to avoid d...	20 hours ago
docs	Update release notes	20 hours ago
docs_src	Add support for declaring <code>UploadFile</code> parameter...	22 days ago
fastapi	Release version 0.73.0	22 days ago
pending_tests	Auto-generate OpenAPI servers from <code>root_path</code> (#1...	2 years ago
scripts	Lint only in Python 3.7 and above (#4006)	4 months ago
tests	Fix bug preventing to use OpenAPI when using tupl...	22 days ago

<https://github.com/tiangolo/fastapi>

Olhando a organização de projetos

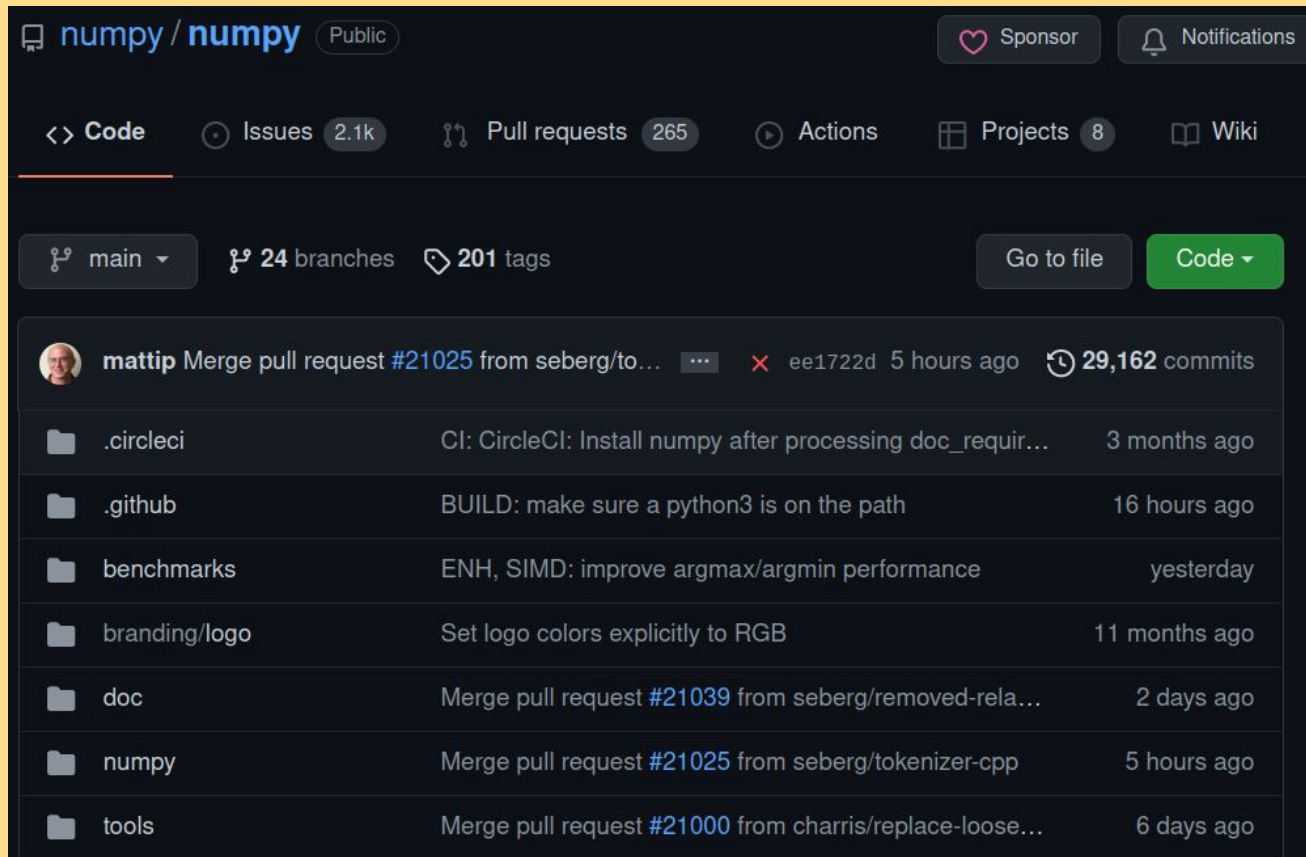


The screenshot shows the GitHub repository page for `python-poetry / poetry`. The repository is public and has a notification bell icon. The navigation bar includes links for Code, Issues (1.1k), Pull requests (165), Discussions, Actions, and Projects. Below the navigation bar, there are buttons for switching branches (currently on `master`), viewing 16 branches, and 108 tags. There are also buttons for "Go to file" and "Code". The commit history section shows a commit by `radoering` titled "Write errors and warnings to stderr (instead of..." with a green checkmark, commit hash `7cc6849`, and a timestamp of "3 days ago". It also shows a total of 2,038 commits. Below the commit history, there is a table of files and their associated commits.

File	Commit Message	Time Ago
<code>.github</code>	Drop python 3.6 support (#5055)	22 days ago
<code>assets</code>	Update README image	4 years ago
<code>docs</code>	Fix grammar (#5098)	11 days ago
<code>src/poetry</code>	Write errors and warnings to stderr (instead of stdout) ...	3 days ago
<code>tests</code>	Write errors and warnings to stderr (instead of stdout) ...	3 days ago

<https://github.com/python-poetry/poetry>

Olhando a organização de projetos




The screenshot shows the GitHub repository for numpy. At the top, there's a navigation bar with 'Code', 'Issues' (2.1k), 'Pull requests' (265), 'Actions', 'Projects' (8), and 'Wiki'. Below this, there's a section for 'main' branch, '24 branches', and '201 tags'. A 'Go to file' button and a 'Code' button are also present. The main content area shows a merge pull request #21025 from seberg/to... by mattip, merged 5 hours ago, with 29,162 commits. Below this, there's a list of recent commits with their descriptions and timestamps.

Commit	Description	Time
.circleci	CI: CircleCI: Install numpy after processing doc_requir...	3 months ago
.github	BUILD: make sure a python3 is on the path	16 hours ago
benchmarks	ENH, SIMD: improve argmax/argmin performance	yesterday
branding/logo	Set logo colors explicitly to RGB	11 months ago
doc	Merge pull request #21039 from seberg/removed-rela...	2 days ago
numpy	Merge pull request #21025 from seberg/tokenizer-cpp	5 hours ago
tools	Merge pull request #21000 from charris/replace-loose...	6 days ago

<https://github.com/numpy/numpy>


Olhando a organização de projetos








 **dunossauro / tigre** Public Notification

<> Code Issues 16 Pull requests 2 Actions Projects Wiki Sec

master 3 branches 0 tags Go to file Code

 **dunossauro** Update gitignore with project stuff ✓ c7a4e03 on 25 Mar 2020 🕒 48 commits

	<a>.github/workflows	Introducing CI	2 years ago
	<a>docker_test_images	docker-selenium compose to example	2 years ago
	<a>docs	Update development docs, for running and register a ...	2 years ago
	<a>tests	Merge branch 'master' of https://github.com/dunossau...	2 years ago
	<a>tigre	Introducing CI	2 years ago

<https://github.com/dunossauro/tigre>

Mas meu projeto é assim



arquivo.py

arquivo2.py

arquivo3.py

O meu projeto

Apresentando nosso projeto de
exemplo



Bora pro código



Ferr ame ntas

Para auxiliar o
desenvolvimento

Escolher a versão do Python



Preferencialmente usar a **ultima possível**

Nem sempre as bibliotecas que dependemos, oferecem suporte a última versão do python, cabe a nós descobrirmos qual a ultima versão suportada por uma biblioteca que dependemos.

Dica de ferramenta: **pyenv** (<https://github.com/pyenv/pyenv>)

Guia básico do pyenv



```
# instale o pyenv - https://github.com/pyenv/pyenv#installation
```

```
pyenv install 3.10.2 # Instala a versão 3.10.2
```

```
pyenv local 3.10.2    # Diz que meu projeto usará 3.10.2
```

```
# Criará o arquivo .python-version
```

Escolha de ferramenta para ambientes virtuais



Aqui existe uma **escolha pessoal**. Algumas pessoas preferem usar virtualenv, venv, poetry, pipenv (ver live #191)

No nosso caso, hoje, usaremos o **poetry**.

Uso básico do poetry



```
# instale o poetry - pipx install poetry

poetry init # Iniciará o projeto
poetry shell # Inicia o ambiente virtual

poetry add <pacote> # instala pacotes
poetry add --dev <pacote> # instala pacote para desenvolvimento
```

Mais informações: Live 179

Versionamento de código



Notou algo em comum com todos os projetos que vimos?

Sim, todos estão no github e usam **GIT** (<https://git-scm.com/>) para fazer controle de versão.

O tutorial tosco de git do mundo



```
# Instale o git - https://git-scm.com/  
# Configure suas credenciais do git  
  
git init      # Inicia o controle de versão  
git add -p .  # Seleciona as mudanças que quer adicionar  
git commit -m "Sua mensagem" # Grava as modificações  
git push      # Coloca as modificações online (github/gitlab)
```

Não podemos esquecer de ignorar



Um dos arquivos importantes para um projeto versionado com git é o arquivo **.gitignore**. Nele dizemos as coisas que não queremos subir para o nosso repositório.

Uma dica: <http://gitignore.io>

Dica para aprender git: <https://gitfichas.com/>

Adicionando testes



Testes são uma peça fundamental de qualquer projeto python. Garantem horas de sanidade mental. Temos diversas ferramentas para isso:

- pytest (live #167)
- unittest (live #75)
- behave (live #10)
- ward
- ...

Utilização básica do pytest



```
— □ ×  
  
# instalação do pytest  
poetry add --dev pytest  
  
# criação da pasta de testes  
mkdir tests  
touch tests/__init__.py  
  
pytest # roda os testes
```

Exemplo de um teste básico



```
from cesar.cesar import encripta
```

```
def test_encripta_eduardo_deve_retornar_rqhneqb():  
    assert encripta('Eduardo') == 'rqhneqb'
```

Formatadores automáticos de código



Já sentiu que quando você programa com outra pessoa as coisas tem uma "estática" um pouco diferente? Como assim?

- Ordem de imports diferentes
- " ou '?
- Quebra de linha de maneiras diferentes
- Tamanho de linhas sem padrão
- ...
-

Formatadores de código



Para buscar uma padronização na maneira de escrever código, temos ferramentas de formatação:

- blue
- black
- isort
- autopep8
- YAPF
- **darker***
- ...

Exemplo do Black em ação

```
# Entrada
def very_important_function(template: str, *variables, file: os.PathLike,
                             engine: str, header: bool = True,
                             debug: bool = False):
    """Applies `variables` to the `template` and writes to `file`."""
    with open(file, 'w') as f: ...
```

```
# Saída
def very_important_function(
    template: str,
    *variables,
    file: os.PathLike,
    engine: str,
    header: bool = True,
    debug: bool = False,
):
    """Applies `variables` to the `template` and writes to `file`."""
    with open(file, "w") as f:
        ...
```

Criadores de documentação



Formatar documentação nem sempre é fácil. Para isso existem ferramentas como:

- mkdocs (live #189)
- sphinx
- ...

Exemplo de uso do mkdocs



```
# instalação do mkdocs  
poetry add --dev mkdocs  
mkdocs new .  
  
# criara a pasta /docs  
# e o arquivo mkdocs.yml
```


Chec
agem

Garantindo a
sanidade do nosso
projeto



Diferente dos testes, as ferramentas de análise estática procuram:

- Erros de sintaxe
- Potenciais problemas
 - Nomes duplicados
 - Nomes ruins
 - Códigos inseguros
- Análise de complexidade de código (Live #38)
- Violações de convenções (Live # 39)
 - PEP-8
 - PEP-257

Ferramentas



Existem diversas ferramentas nessa linha, algumas recomendações:

- **flake8**: Checador da pep-8
- **pylint**: Padronização, convenções e erros
- **pydocstyle**: Checador da PEP-257
- **bandit**: Problemas de segurança
- **Radon**: Busca de complexidade de código
- **Prospector**: Agregador de ferramentas
 - Flake8, McCabe, pylint, pep8, pep257, ...
- **mypy**: Checador de sugestões de tipo
- ...

Usando o prospector



```
poetry add --dev prospector # Instala  
prospector # Roda as analises  
  
# Executa também a chagem de docstrings  
prospector --with-tool pep257
```

Segurança de bibliotecas



Com o passar do tempo, às vezes algumas bibliotecas ficam "congeladas" no nosso projeto. E isso pode trazer várias vulnerabilidades de segurança (Live #163). Algumas ferramentas podem nos ajudar:

- pip-audit
- safety
- pyup
- ...

Exemplo de uso pip-audit



```
poetry add --dev pip-audit # Instala
```

```
pip-audit # Roda as analises
```

Exemplo de quando existe problema



```
$ poetry add flask=0.12 # versão desatualizada do Flask
```

```
$ pip-audit
```

```
Found 2 known vulnerabilities in 1 packages
```

Name	Version	ID	Fix Versions
------	---------	----	--------------

flask	0.12	PYSEC-2019-179	1.0
flask	0.12	PYSEC-2018-66	0.12.3

Name	Skip Reason
------	-------------

--	--

Facilitando a vida e
os comandos

Autom
ações

MUITOS, MAS MUITOOS comandos



black, isort, poetry, prospector, pip-audit, pytest, mkdocs ...

Bom, isso fica confuso de MAIS, vamos criar comandos mais simples?

GNU Make

live #184



```
# Makefile
.PHONY: install format lint test sec

install:
    @poetry install
format:
    @blue .
    @isort .
lint:
    @blue --check .
    @isort --check .
    @prospector
test:
    @pytest -v
sec:
    @pip-audit
```

A paz de não esquecer nada



Comandos mais simples, fazem como que nós executemos mais os comandos, mas tem o "fator humano". As vezes a gente simplesmente esquece. Podemos usar os **hooks** do **git** para rodar as coisas.

- Antes do commit
- Antes do rebase
- Antes do push
- Antes do merge
- ...

Configurando os Hooks



```
# .git/hooks/pre-commit
# Esse arquivo precisa ser executável
# chmod +x (no windows usa o git shell)
make lint
```

Quando as coisas dão errado no git



Nem sempre tudo são flores, o git faz muita coisa no servidor (github/gitlab) o ideal seria ter uma maneira de fazer essas checagens no repositório também. Para isso temos o Actions (Live #170)

Github Actions

```
# .github/workflows/continuous_integration.yml
```

```
name: Continuous Integration
```

```
on: [push]
```

```
jobs:
```

```
  lint_and_test:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

- name: Set up python
 uses: actions/setup-python@v2
 with:
 python-version: 3.10
- name: Check out repository
 uses: actions/checkout@v2
- name: Install Poetry
 uses: snok/install-poetry@v1
 with:
 virtualenvs-in-project: true

```
# .github/workflows/continuous_integration.yml (continuação)
```

- name: Load cached venv
 id: cached-poetry-dependencies
 uses: actions/cache@v2
 with:
 path: .venv
 key: venv-\${{ hashFiles('**/poetry.lock') }}
- name: Install dependencies
 if: steps.cached-poetry-dependencies.outputs.cache-hit != 'true'
 run: poetry install --no-interaction
- name: Lint
 run: poetry run make lint
- name: Run tests
 run: poetry run make test



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3

