



Poetry

Live de Python # 179



1. Pacotes

Uma história, uma odisseia

2. Projeto

Conhecendo o poetry

3. Pacote

Empacotando nossa aplicação

4. Distribuição

Distribuindo a nossa aplicação <3



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto



Acássio Anjos, Ademar Peixoto, Alex Lima, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alexandre Villares, Alynne Ferreira, Alysso Oliveira, Amaziles Carvalho, Andre Azevedo, André Rocha, Antonio Lins, Arnaldo Turque, Artur Zalewska, Bruno Barcellos, Bruno Batista, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Caio Nascimento, Carlos Chiarelli, Cleber Santos, César Almeida, Davi Ramos, David Kwast, Diego Guimarães, Diego Ubirajara, Dilenon Delfino, Dino Aguilar, Donivaldo Sarzi, Elias Soares, Emerson Rafael, Eric Niens, Eugenio Mazzini, Euripedes Borges, Everton Alves, Fabiano Gomes, Fabio Barros, Fabio Castro, Flavkaze Flavkaze, Flávio Meira, Francisco Alencar, Franklin Silva, Fábio Barros, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Castro, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Suto, Henrique Junqueira, Henrique Machado, Ismael Ventura, Israel Fabiano, Israel Gomes, Italo Silva, Jair Andrade, Jairo Rocha, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, Jose Mazolini, José Gomes, José Prado, João Lugão, Juan Gutierrez, Jônatas Silva, Kaio Peixoto, Kaneson Alves, Leonardo Cruz, Leonardo Mello, Leonardo Nascimento, Lidiane Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Luciano Silva, Maiquel Leonel, Marcela Campos, Marcelino Pinheiro, Marcos Ferreira, Maria Clara, Marina Passos, Matheus Vian, Murilo Cunha, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Patric Lacouth, Patricia Minamizawa, Patrick Brito, Patrick Gomes, Paulo Tadei, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Renan Moura, Revton Silva, Richard Nixon, Riverfount Riverfount, Robson Maciel, Rodrigo Ferreira, Rodrigo Mende, Rodrigo Vaccari, Rodrigo Vieira, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sandro Mio, Sara Selis, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Moraes, Tony Dias, Tony Santos, Tyrone Damasceno, Vinícius Bastos, Vlademir Souza, Vítor Gomes, Wellington Abreu, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte, Yuri Fialho, Yury Barros, Érico Andrei



Obrigado você



Pacotes

Usando pacotes

Workflow dos pacotes



Seu código

Workflow dos pacotes



Seu código

virtualenv

```
python -m venv ./venv
```

Workflow dos pacotes



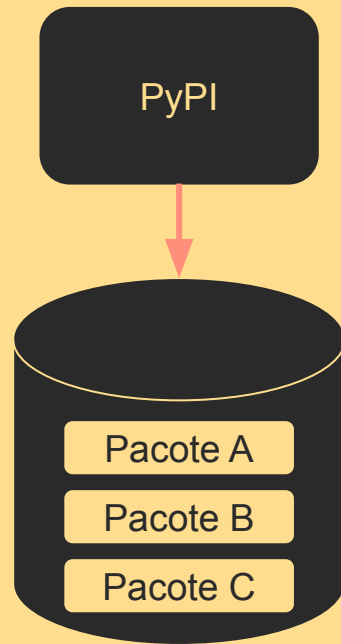
Seu código



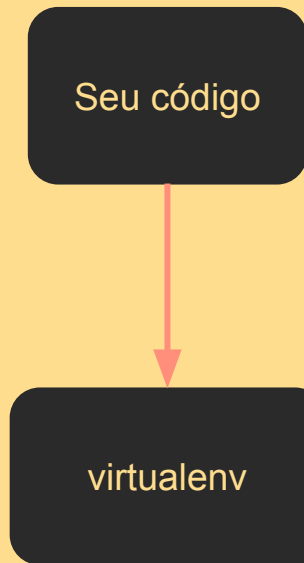
virtualenv

```
python -m venv ./venv  
source ./venv/bin/activate
```

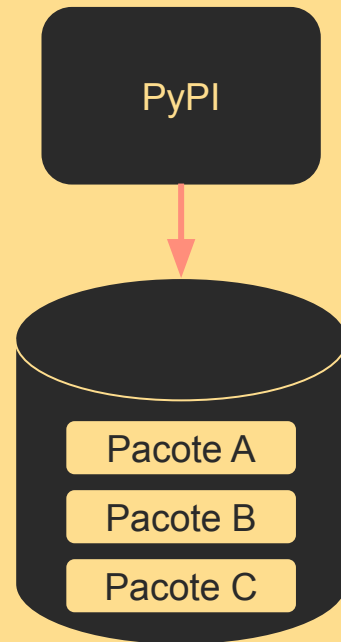

Workflow dos pacotes



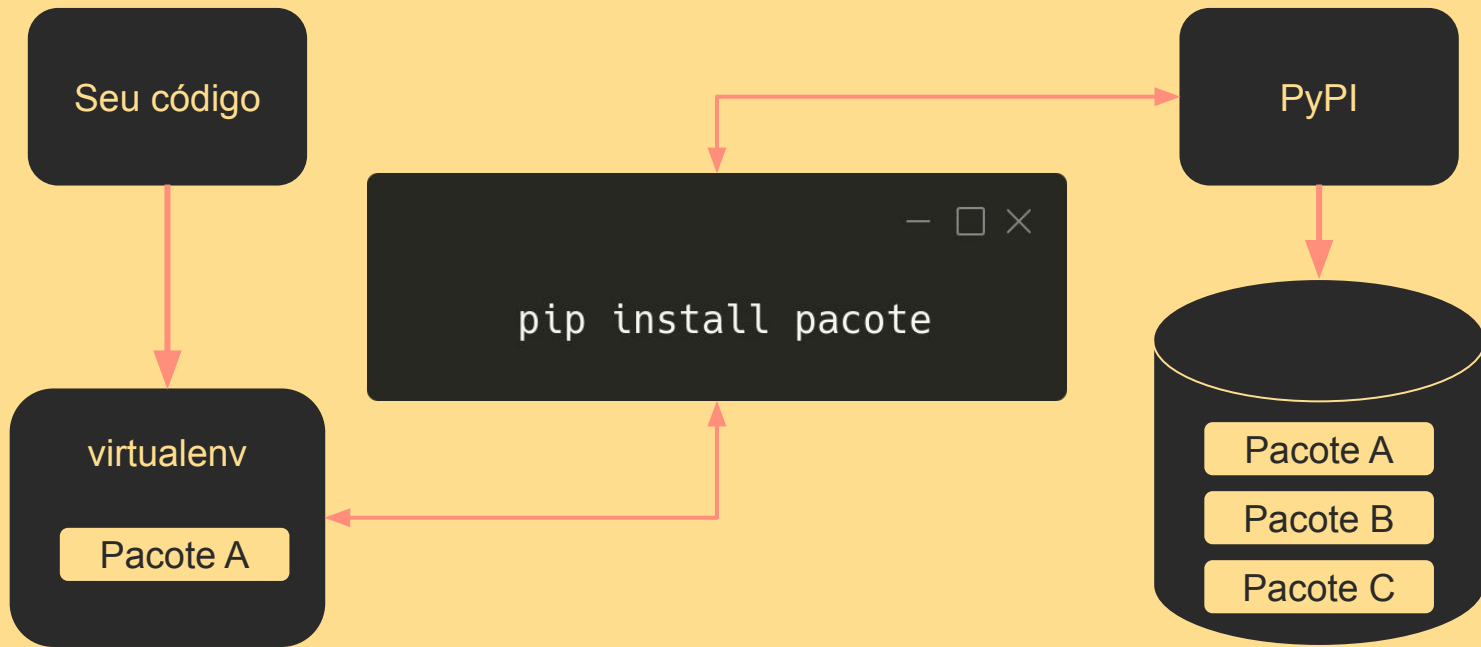
Workflow dos pacotes



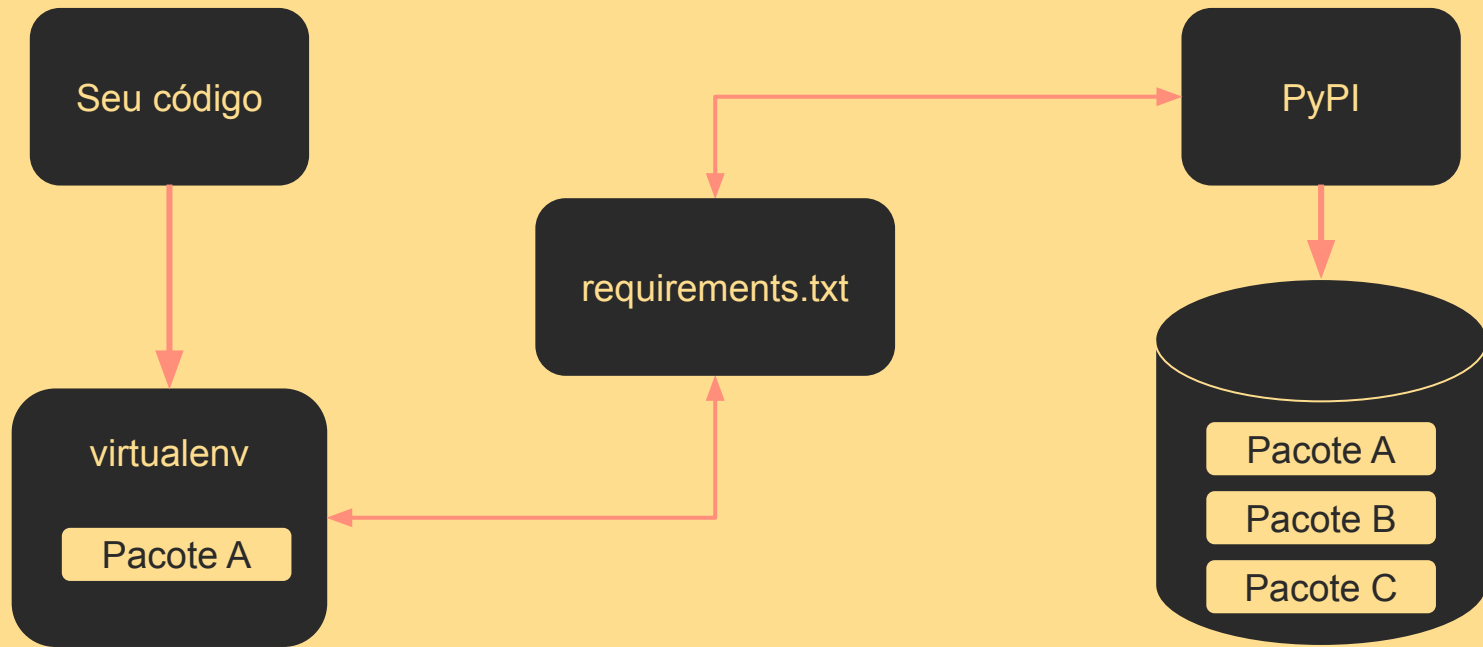
?



Workflow dos pacotes



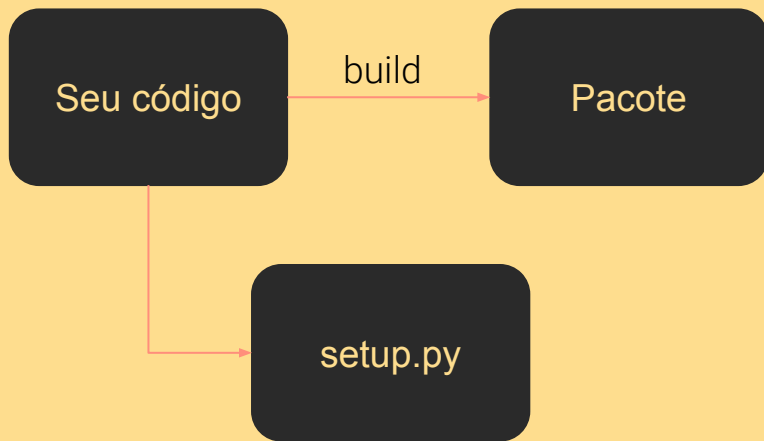
Workflow dos pacotes



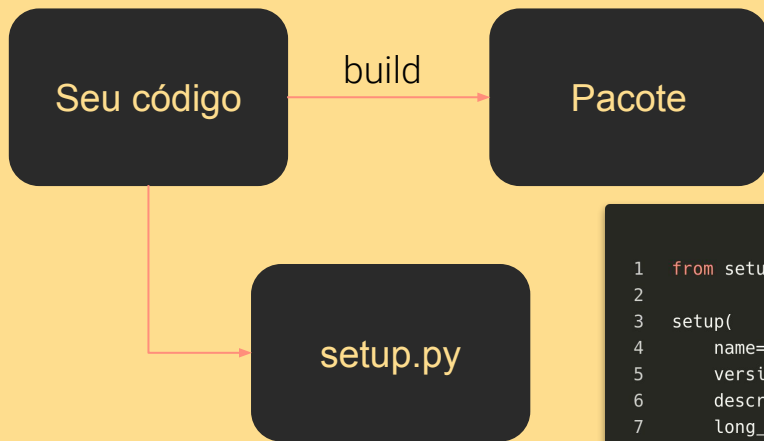
Pacotes

Criando pacotes

Workflow dos pacotes

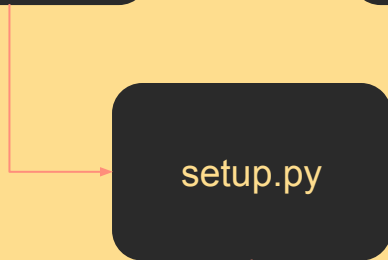


Workflow dos pacotes



```
1  from setuptools import setup
2
3  setup(
4      name='projeto',
5      version='0.1.0',
6      description='Meu projeto',
7      long_description='Meu super projeto, por favor instale!',
8      author='dunossauro',
9      author_email='mendesxeducardo@gmail.com',
10     packages=['projeto'],
11     python_requires='>=3.9,<4.0',
12     entry_points={'console_scripts': ['meu-script = projeto.cli:cli']},
13     install_requires=['httpx'],
14 )
```

Workflow dos pacotes

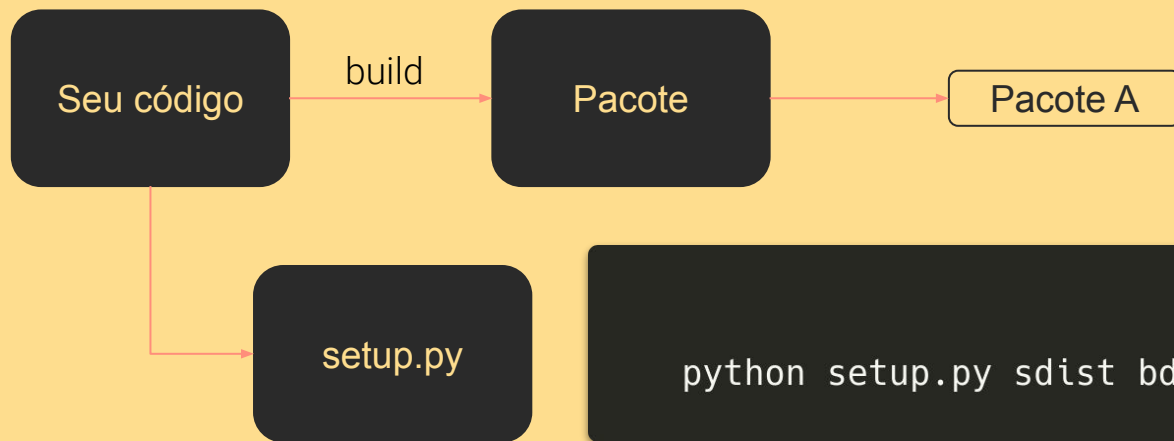


```
1 from setuptools import setup
2
3 setup(
4     name='projeto',
5     version='0.1.0',
6     description='Meu projeto',
7     long_description='Meu super projeto, por favor instale!',
8     author='dunossauro',
9     author_email='mendesxeduardo@gmail.com',
10    packages=['projeto'],
11    python_requires='>=3.9,<4.0',
12    entry_points={'console_scripts': ['meu-script = projeto.cli:cli']}},
13    install_requires=['httpx'],
14 )
```



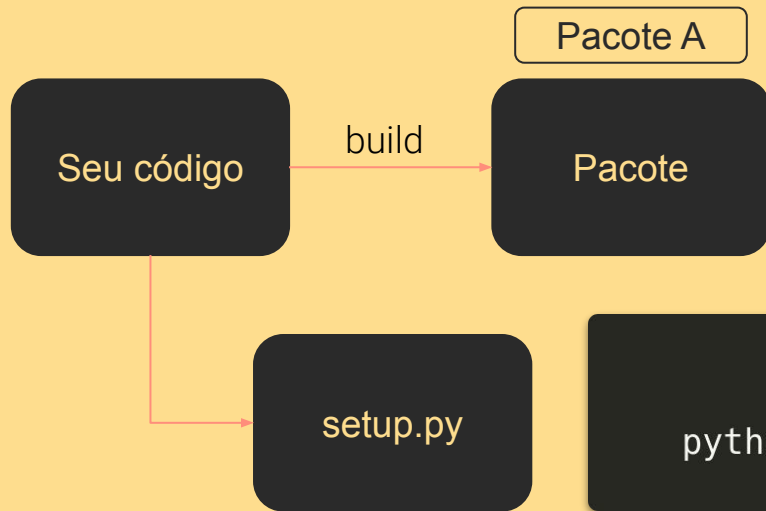
```
1  from setuptools import setup
2
3  setup(
4      name='projeto',
5      version='0.1.0',
6      description='Meu projeto',
7      long_description='Meu super projeto, por favor instale!',
8      author='dunossauro',
9      author_email='mendesxeduardo@gmail.com',
10     packages=['projeto'],
11     python_requires='>=3.9,<4.0',
12     entry_points={'console_scripts': ['meu-script = projeto.cli:cli']},
13     install_requires=['httpx'],
14 )
```

Workflow dos pacotes



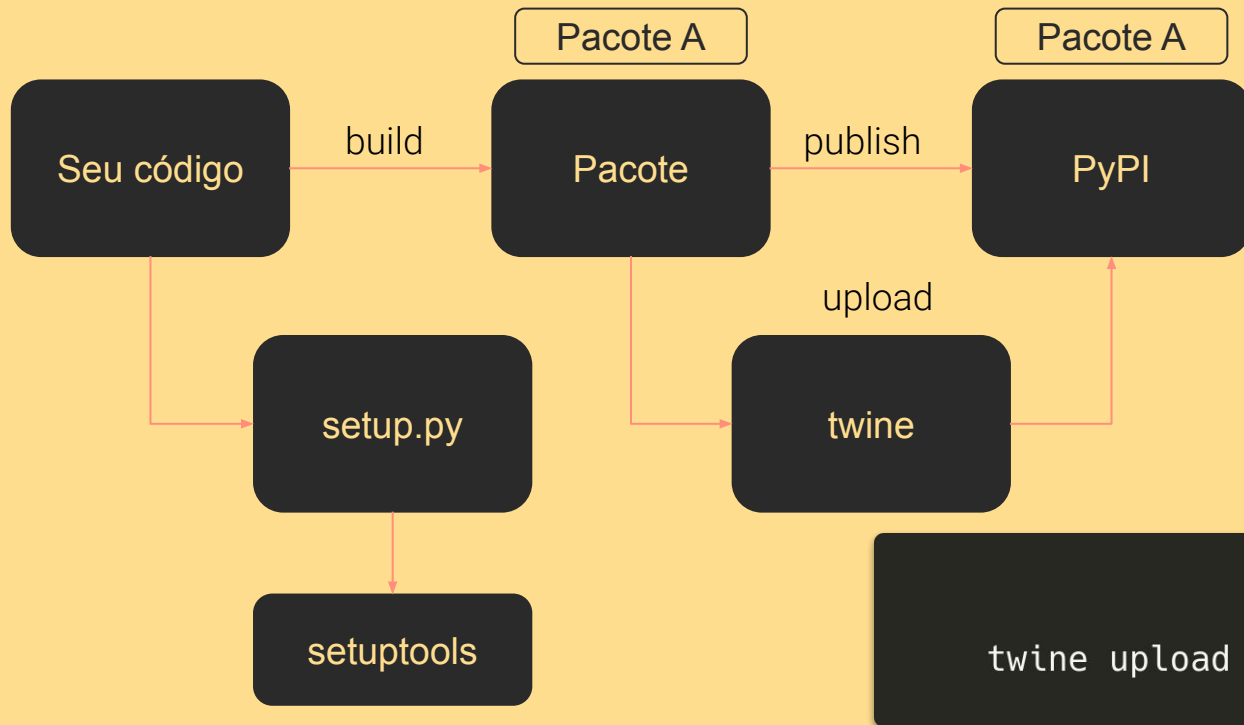
```
python setup.py sdist bdist_wheel
```

Workflow dos pacotes



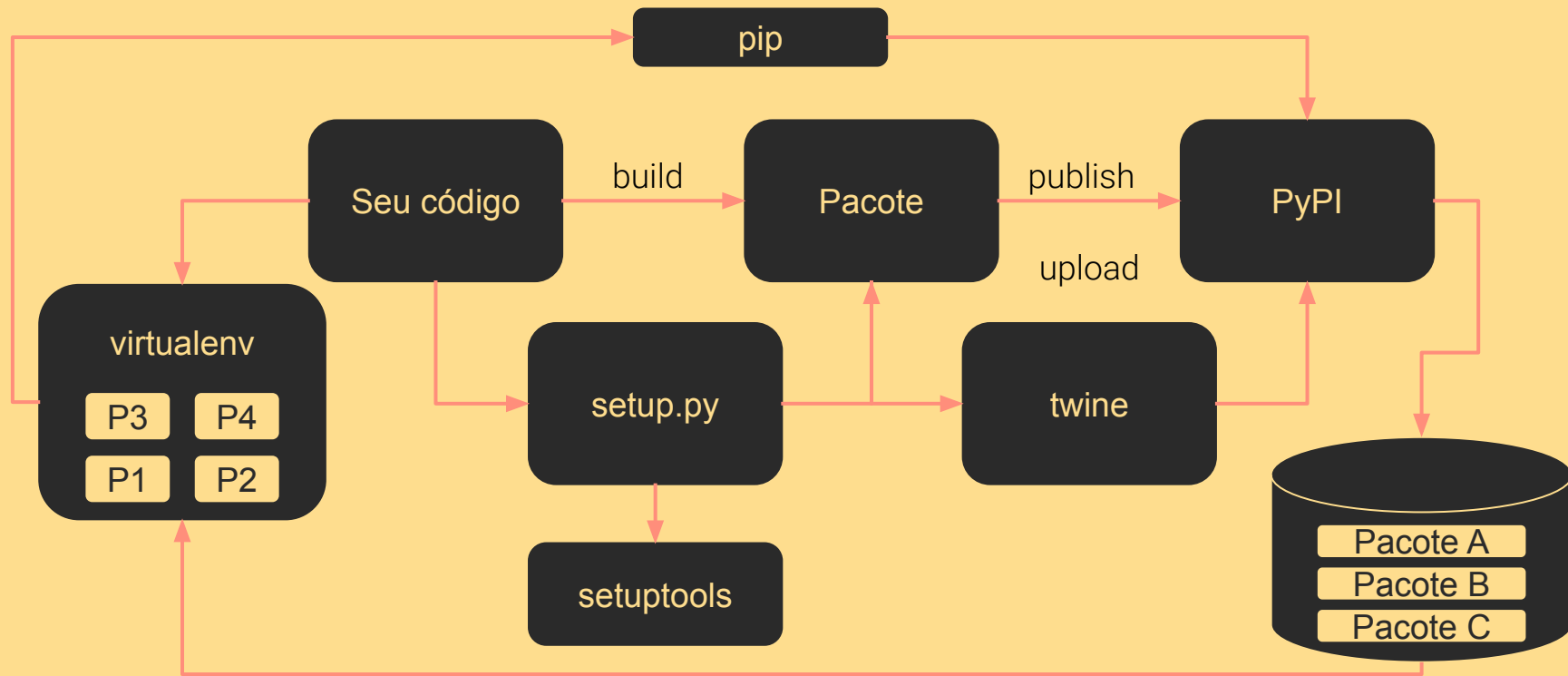
```
python setup.py sdist bdist_wheel
```

Workflow dos pacotes



```
twine upload dist/*
```

Workflow dos pacotes



Uma história



Antiguidade	Idade média	Era moderna
1990 - Disquetes?	2011 - PyPA	2016 - Pyproject.toml
2000 - Distutils	2012 - Conda	2017 - Independent Builds
2003 - PyPi	2012 - pip-tools	2017 - Pipenv
2004 - Setuptools	2013 - Wheel	2018 - Poetry
2007 - Virtualenv	2014 - PEP-440 (sysver)	2019 - novo pip
2008 - Pip	2015 - PyPi Standards	2021 - Essa live

https://www.youtube.com/watch?v=QX_Nhu1zhlg

Um futuro



- PEP-582?
 - node_modules?
 - pdm
- PEP-621?
 - Metadados dos pacotes
 - poetry
 - flit

Projeto

Usando o poetry



Sobre o Poetry



Poetry é uma biblioteca para gerenciamento de dependências e empacotamento para Python.

Criado por Sébastien Eustace em 2018 sob licença MIT.

Instalação

Linux / OS X

```
curl -sSL https://raw.githubusercontent.com/python-poetry/poetry/master/get-poetry.py | python -
```

Windows

```
(Invoke-WebRequest -Uri https://raw.githubusercontent.com/python-poetry/poetry/master/get-poetry.py  
-UseBasicParsing).Content | python -
```

Não dá pra usar pip?



Dá pra instalar via pip também. (pip install poetry)

Porém a instalação via shell faz **vendoring**

Começando do começo



Vamos criar o projeto usando Poetry

```
poetry new projeto
```

Começando do começo



Vamos criar o projeto usando Poetry

```
poetry new projeto
```

```
$ exa --tree
```

```
.
├── pacote
│   ├── pacote
│   │   └── __init__.py
│   ├── pyproject.toml
│   ├── README.rst
│   └── tests
│       ├── __init__.py
│       └── test_pacote.py
```

Começando do começo



Vamos criar o projeto usando Poetry

```
poetry new projeto
```

```
$ exa --tree
```

```
.  
├── pacote  
│   ├── pacote  
│   │   └── __init__.py  
│   ├── pyproject.toml  
│   ├── README.rst  
│   └── tests  
│       ├── __init__.py  
│       └── test_pacote.py
```

Começando do começo



Vamos criar o projeto usando Poetry

```
poetry new projeto
```

```
$ exa --tree
```

```
.
├── pacote
│   ├── pacote
│   │   └── __init__.py
│   ├── pyproject.toml
│   ├── README.rst
│   └── tests
│       ├── __init__.py
│       └── test_pacote.py
```


Começando do começo



Vamos criar o projeto usando Poetry

PEP-508 / PEP-631

Especificação de um sistema mínimo de
build e requisitos do projeto

```
$ exa --tree
```

```
.  
├─ pacote  
│   ├── pacote  
│   │   └─ __init__.py  
│   └─ pyproject.toml  
├─ README.rst  
└─ tests  
    ├── __init__.py  
    └─ test_pacote.py
```

pyproject.toml

```
[tool.poetry]
name = "pacote"
version = "0.1.0"
description = ""
authors = ["dunossauro <mendesxeduardo@gmail.com>"]

[tool.poetry.dependencies]
python = "^3.9"

[tool.poetry.dev-dependencies]
pytest = "^5.2"

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"
```

```
$ exa --tree
```

```
.
├── pacote
│   ├── pacote
│   │   └── __init__.py
│   └── pyproject.toml
├── README.rst
└── tests
    ├── __init__.py
    └── test_pacote.py
```

pyproject.toml



setup.py

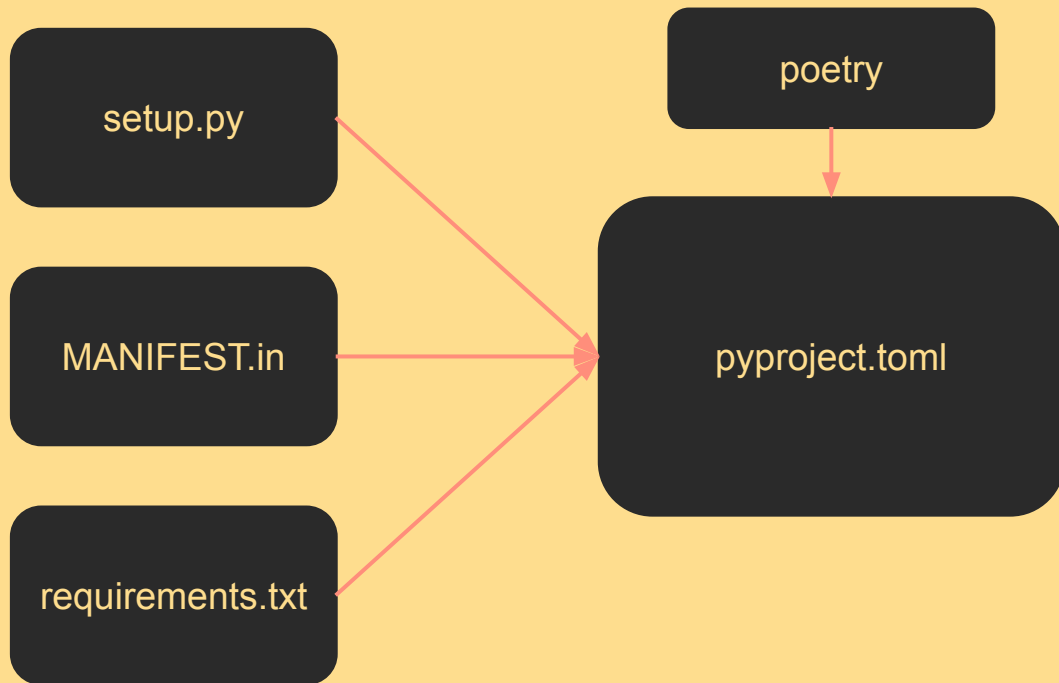
MANIFEST.in

requirements.txt

pyproject.toml



pyproject.toml



Instalar pacotes



```
poetry add httpx
```

Instalar pacotes de desenvolvimento



```
poetry add --dev httpx
```

Criando um CLI



```
# pyproject.toml  
[tool.poetry.scripts]  
meu-script = "projeto.cli:cli"
```

Criando o código



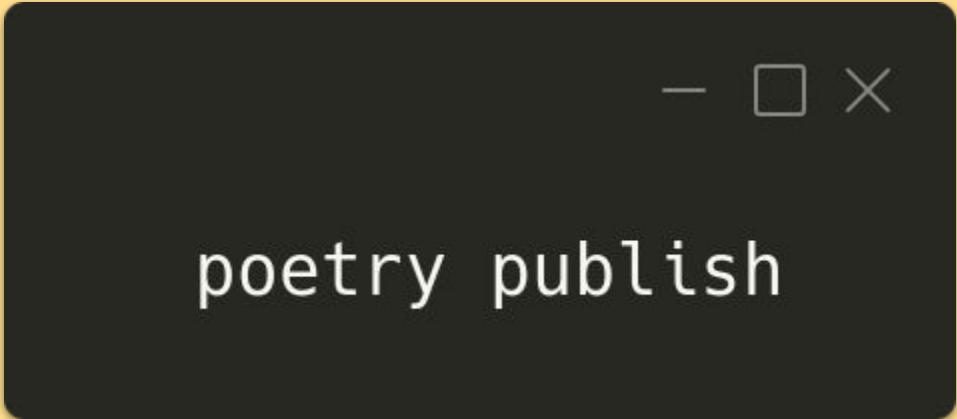
```
1  # projeto/cli.py
2  from httpx import get
3
4
5  def cli():
6      print(
7          get(
8              'http://httpbin.org/get?arg=Live%20de%20Python'
9          ).json()[ 'args' ][ 'arg' ]
10     )
```


Fazendo o build

A dark-themed terminal window with a title bar containing minimize, maximize, and close buttons. The text 'poetry build' is centered within the terminal.

```
poetry build
```

Fazendo a publicação

A dark-themed terminal window with a title bar containing a minus sign, a square icon, and a close 'X' icon. The text 'poetry publish' is centered within the terminal.

```
poetry publish
```



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

