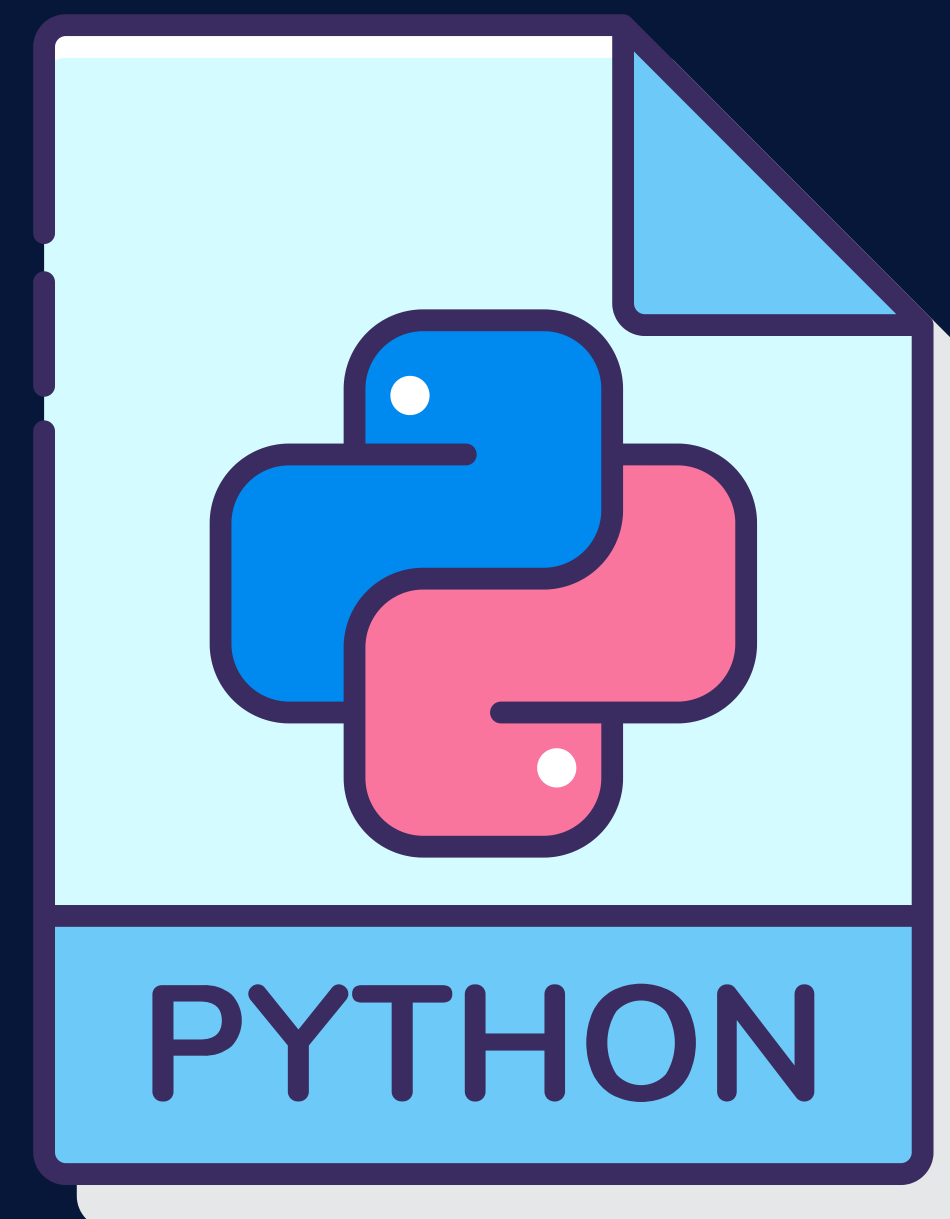


Evento  
Encontro de Engenharia I

# Estruturação

De Projetos de Python do Zero

Por Iury Rosal



# Sumário

1. Problemática

2. Instalação

3. Gerenciamento de versões

4. Trabalhando com pacotes

5. Um novo jeito de instalar pacotes

6. Unindo tudo em uma única solução

7. Estrutura de Pastas

8. Conectando as coisas

Acesse o repositório do projeto:

<https://github.com/iuryrosal/apoenastack>



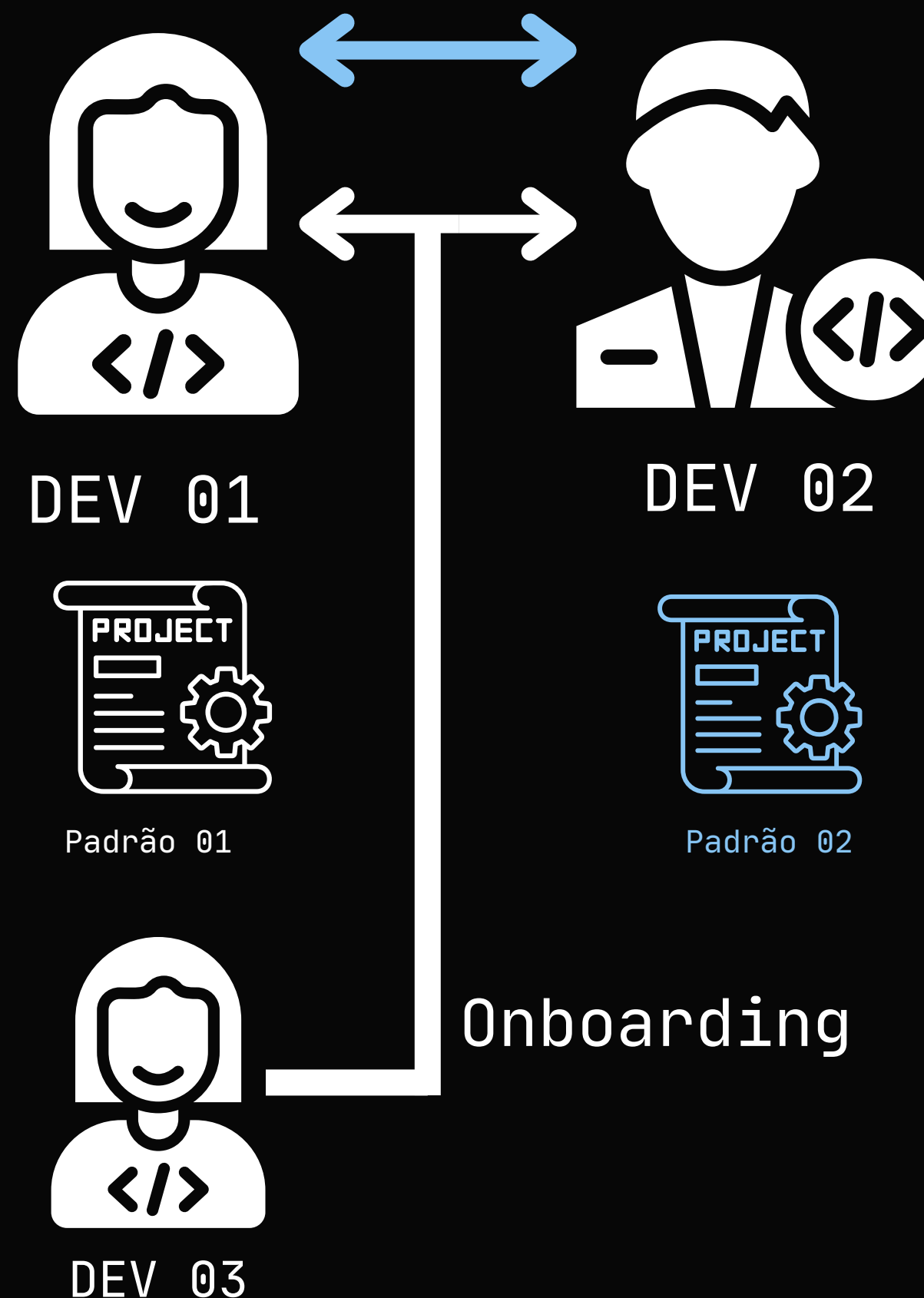
[Entre no grupo do zap sobre Engenharia de Dados!](#)

<https://apoenastack.com>

# O BOSS DE HOJE!



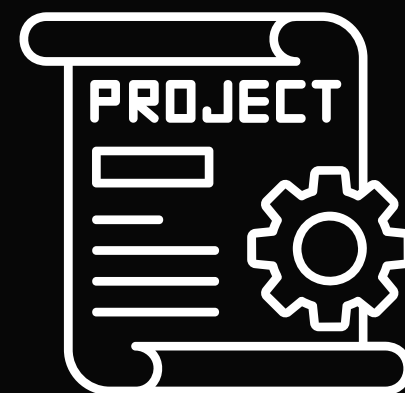
Bagunça do Projeto  
Dificuldade de onboarding  
Dificuldade de manutenção



# O BOSS DE HOJE!

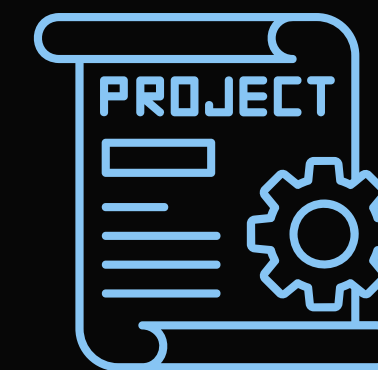


Bagunça do Projeto  
Dificuldade de onboarding  
Dificuldade de manutenção



Projeto 01

Lib01 v. 1.3  
Lib03 v. 1.4  
Lib05 v. 4.8  
Python v. 3.10



Projeto 02

Lib01 v. 1.2  
Lib03 v. 1.8  
Lib05 v. 4.8  
Python v. 3.9

# Instalação de Ferramentas

- WSL
- Python
- Visual Studio Code



# WSL 2

## Definição

O WSL permite que você execute um ambiente de linha de comando GNU/Linux integrado diretamente com o Windows e suas ferramentas favoritas, como Visual Studio Code, Outlook etc.

<https://learn.microsoft.com/pt-br/windows/wsl/install>

[https://medium.com/@francisco\\_51376/python-wsl-2-75e38ab368c9](https://medium.com/@francisco_51376/python-wsl-2-75e38ab368c9)

# Instalação do Python e Visual Studio Code

<https://www.python.org/downloads/>

<https://code.visualstudio.com>

# Extensões do Visual Studio Code

<https://marketplace.visualstudio.com/items?itemName=ms-python.python>

<https://marketplace.visualstudio.com/items?itemName=ms-python.vscode-pylance>

<https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl>

<https://marketplace.visualstudio.com/items?itemName=ms-python.flake8>



# Gerenciamento de versões



# 1º FASE DO BOSS...



Seu computador

Python 3.10

Python 3.11



Projeto1

python 3.10



Projeto2

python 3.11

# PyEnv

## Instalar uma nova versão do Python



```
pyenv install <version>
```

## Ver versões



```
pyenv versions
```

# PyEnv

## Definir uma versão global

```
pyenv global <version>
```

# Trabalhando com Pacotes

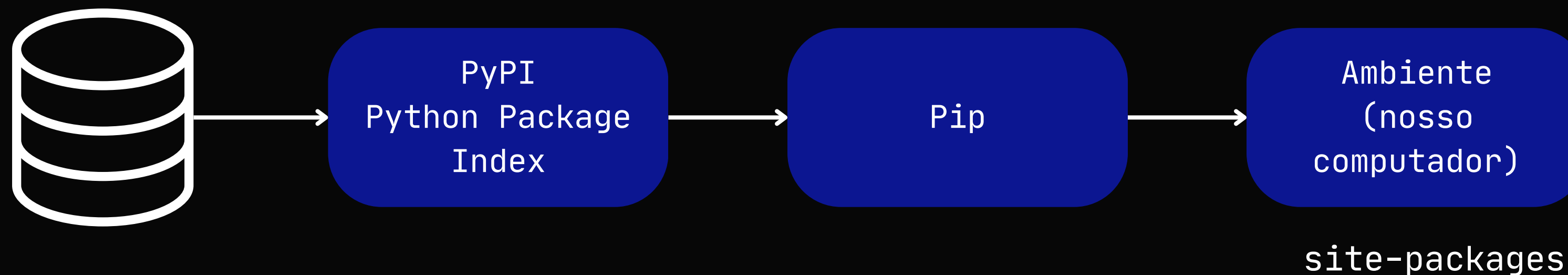
- pip
- venv
- requirements.txt
- setup.py



# Pip

## Definição

É uma maneira de instalar pacotes externos no nosso ambiente.



# Pip

## Instale um pacote



```
pip install <nome-do-pacote>
```

## Veja pacotes instalados



```
pip list
```

## Desinstale um pacote



```
pip uninstall <nome-do-pacote>
```

# Pip

- Explore as dependências:

## Verificar dependências

```
pip show pandas
```

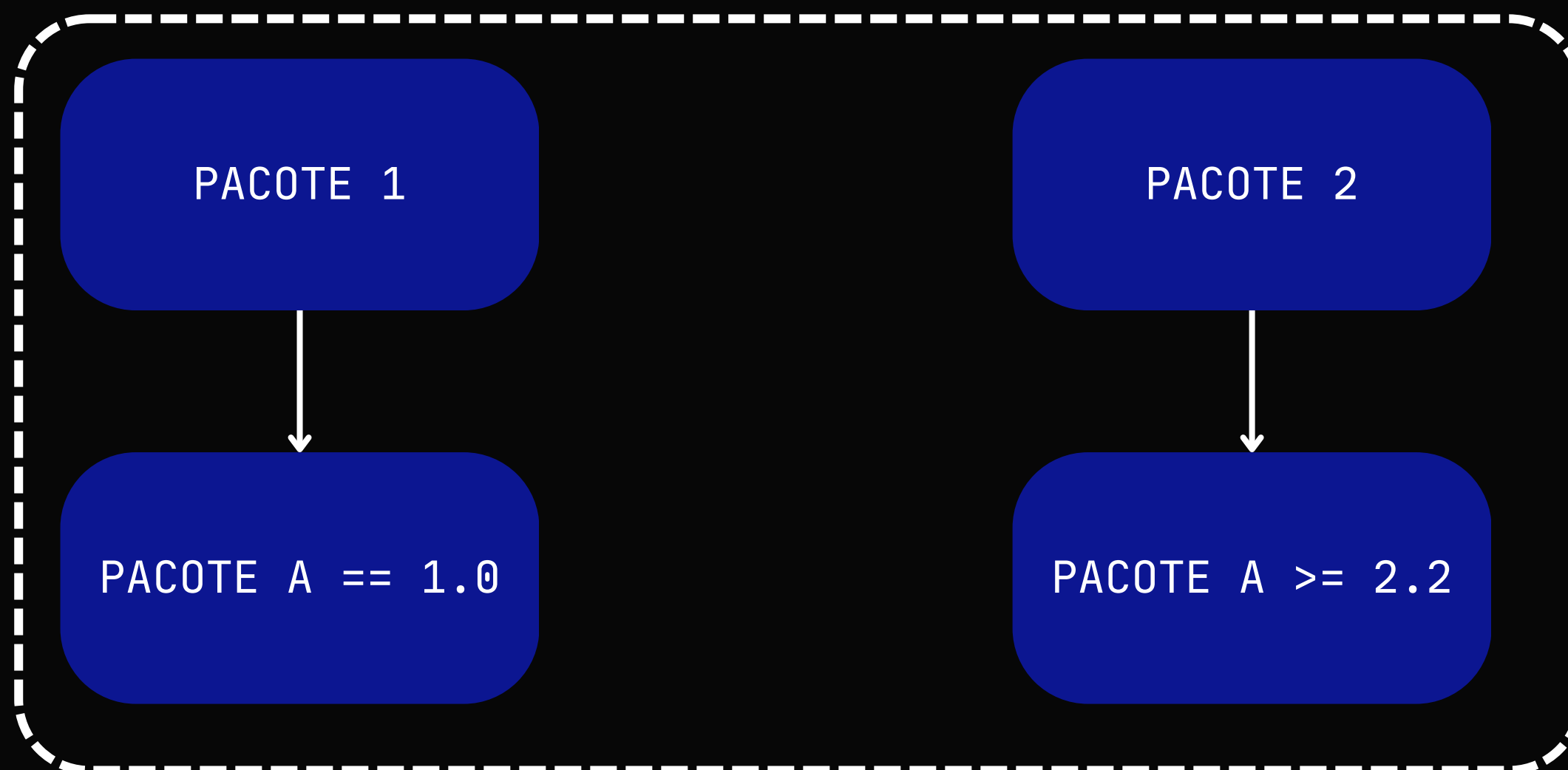
## Remover pacote com dependências

```
pip-autoremove pandas
```



# Pip

- Não é possível ter pacotes na sua máquina com versões diferentes simultaneamente.
- Risco de conflito interno:



# VENV

- Uma possibilidade de 'dibrar' o site-packages
- Cria um ambiente local, isolado do global;
- É diferente de VIRTUALENV (hoje, VENV compõe uma parte do VIRTUALENV).

## Criação do Ambiente Virtual

```
python -m venv <nome_do_ambiente>
```

# Ativando ambiente virtual

Platform	Shell	Command to activate virtual environment
POSIX	bash/zsh	<code>\$ source &lt;venv&gt;/bin/activate</code>
	fish	<code>\$ source &lt;venv&gt;/bin/activate.fish</code>
	csh/tcsh	<code>\$ source &lt;venv&gt;/bin/activate.csh</code>
	PowerShell	<code>\$ &lt;venv&gt;/bin/Activate.ps1</code>
Windows	cmd.exe	<code>C:\&gt; &lt;venv&gt;\Scripts\activate.bat</code>
	PowerShell	<code>PS C:\&gt; &lt;venv&gt;\Scripts\Activate.ps1</code>

<https://docs.python.org/3/library/venv.html#creating-virtual-environments>

## Desativar ambiente

deactivate

# requirements.txt

- Arquivo que lista os pacotes e versões que são utilizadas;
- Facilita a preparação de ambiente e transportar seu projeto para outras localidades.

```
requirements.txt
1  # local package
2  -e .
3
4  # external requirements
5  Sphinx
6  coverage
7  awscli
8  flake8
9  python-dotenv>=0.5.1
10 matplotlib==3.3.3
11 numpy==1.19.4
12 pandas==1.1.5
13 pandas-profiling==3.6.6
14 seaborn==0.11.1
15 streamlit==1.8.0
16 plotly
17 tk
18 click<=8.0.4
19 scikit-learn
```

Apenas nome do pacote: pip busca a versão mais atualizada (não recomendado)

É possível especificar as versões: exatamente a versão (`==`), antes de uma versão (`<` / `<=`) ou depois de uma versão (`>` / `>=`).

Use o site do [PyPI](#) para ver todas as versões dos pacotes

# requirements.txt

- Arquivo que lista os pacotes e versões que são utilizadas;
- Facilita a preparação de ambiente e transportar seu projeto para outras localidades.

## Instalar pacotes usando o requirements.txt

```
pip install -r requirements.txt
```

# requirements.txt

- Use comandos `pip` para montar o arquivo

## Instalar pacotes usando o requirements.txt

```
pip freeze > requirements.txt
```

# requirements.txt

- Use comandos `pip` para montar o arquivo

## Instalar pacotes usando o requirements.txt

```
pip freeze > requirements.txt
```



Nem todas as bibliotecas são necessárias (algumas são dependências de libs principais que já são instaladas baseado no pacote de origem e que podem variar dependendo da versão)

**Solução:** Pode usar o freeze e remover depois as libs que não são as principais

# requirements.txt

- Use comandos `pip` para montar o arquivo

## Instalar pacotes usando o requirements.txt

```
pip freeze > requirements.txt
```



Nem todas as bibliotecas são necessárias (algumas são dependências de libs principais que já são instaladas baseado no pacote de origem e que podem variar dependendo da versão)

**Solução:** Pode usar o freeze e remover depois as libs que não são as principais



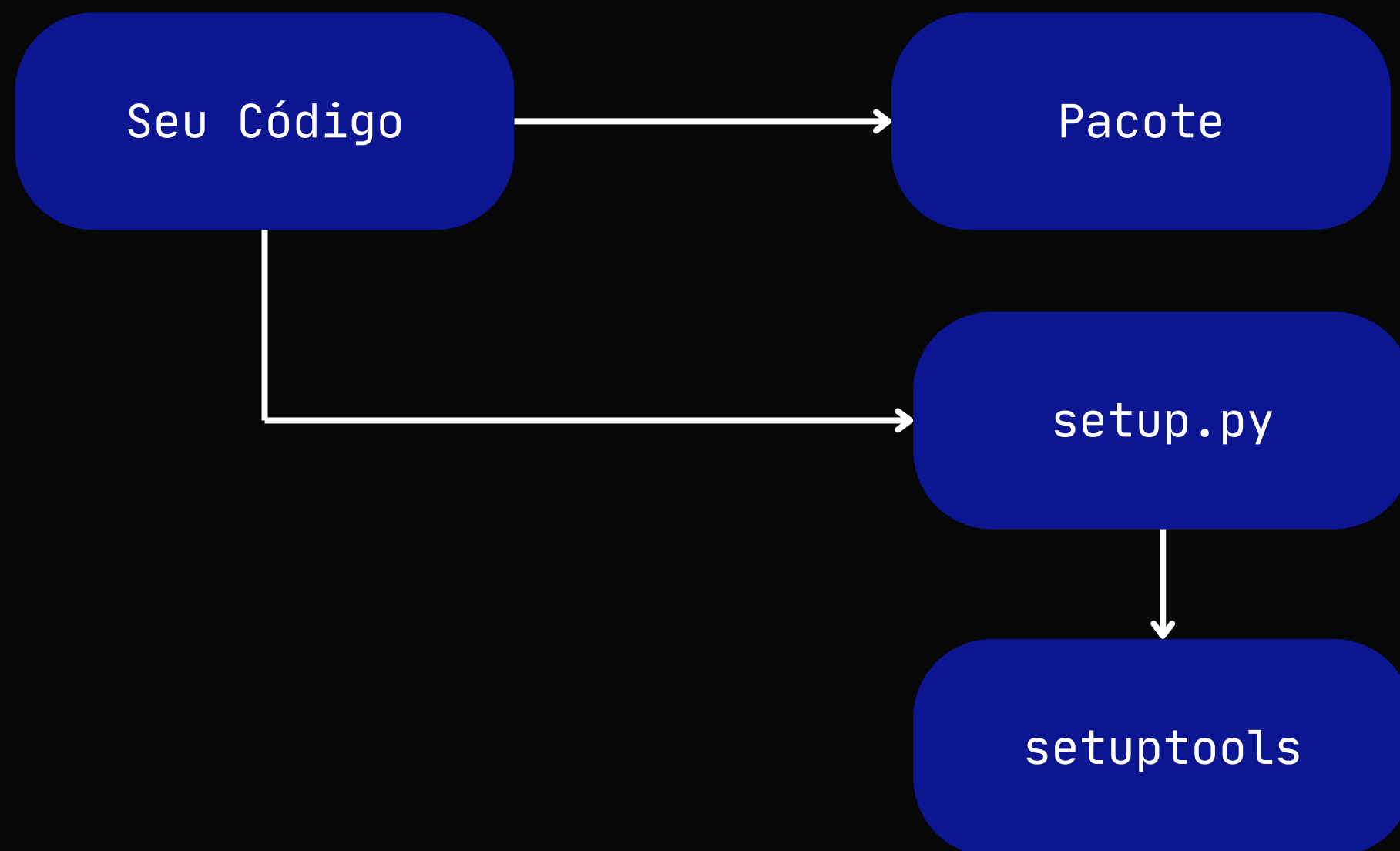
Algumas libs são apenas de interesse para o ambiente de desenvolvimento, não o de produção....

**Solução:** Criar uma requirements-dex.txt



# setup.py

- Realiza o build do nosso projeto, ou seja, irá gerar o pacote do nosso projeto, facilitando mais ainda o processo de reutilização do pacote.



```
setup.py
1  from setuptools import find_packages, setup
2
3  setup(
4      name='src',
5      version='1.0',
6      description='Projeto para o TCC',
7      author='Iury Rosal',
8      author_email='foomail@foo.example',
9      packages=find_packages(),
10     license=''
11 )
```

# Um novo jeito de instalar pacotes

- PipX



# PipX

- Instala ferramentas de linha de comando em um ambiente virtual isolado. Evita sujar o ambiente global!

## Instalar o pipx pelo pip

```
pip install pipx
```

## Instalar ferramentas usando o pipx

```
pipx install <pacote>
```



PIP



VENV



PIPX



PYENV



POETRY

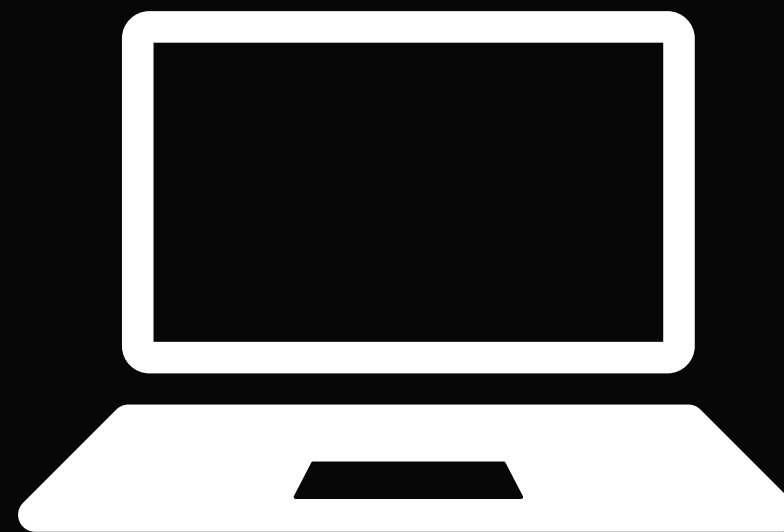
# Unindo tudo em uma única solução

- Poetry



# FASE FINAL DO BOSS..

pandas 1.5, 2.0, 2.1, 2.2

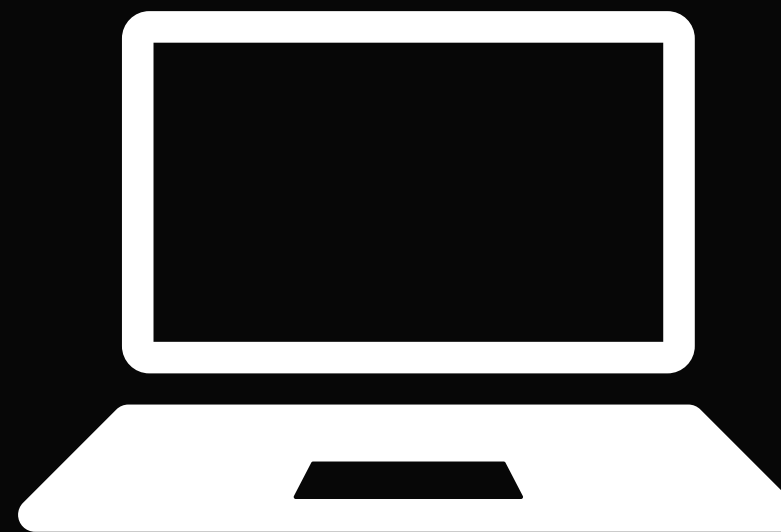


Seu computador

Seu computador tem a  
versão 2.2 do pandas por  
conta de outro projeto

# MAIS UMA FASE DO BOSS..

`pandas 1.5, 2.0, 2.1, 2.2`



Seu computador

Seu computador tem a  
versão 2.2 do pandas por  
conta de outro projeto



Projeto1

`pandas 1.5`

`feature1`

Disponível apenas na 1.5



Projeto2

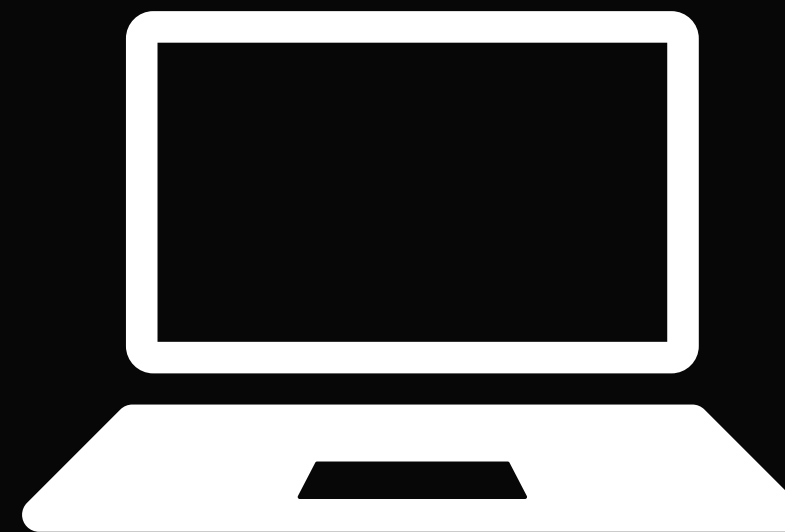
`pandas 2.1`

`feature2`

Disponível apenas a partir da 2.1

# MAIS UMA FASE DO BOSS..

pandas 1.5, 2.0, 2.1, 2.2



Seu computador

Seu computador tem a  
versão 2.2 do pandas por  
conta de outro projeto



Projeto1

pandas 1.5

feature1

Disponível apenas na 1.5



Projeto2

pandas 2.1

feature2

Disponível apenas a partir da 2.1

MULTIPLICA ISSO POR 50...



# Poetry

## Definição

**Biblioteca para gerenciamento de dependências e empacotamento para Python**

## Instalar pacotes



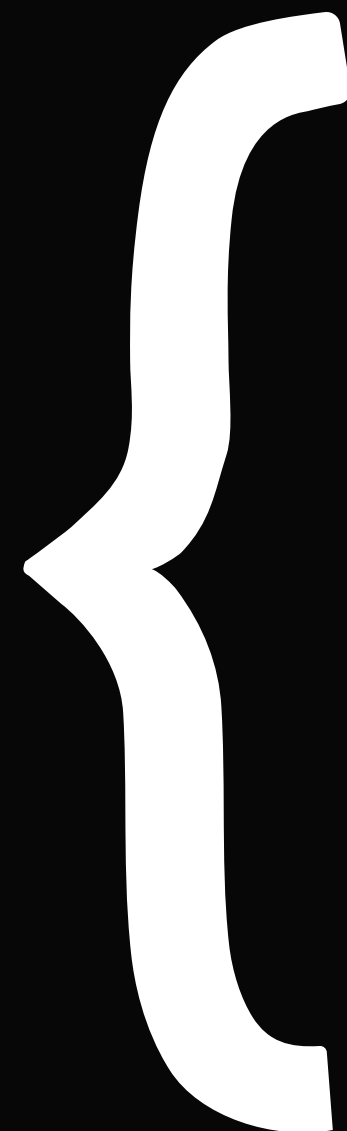
```
poetry new poetry-demo
```

# Poetry

setup.py

requirements.txt

venv



pyproject.toml

# Poetry

## Instalar pacotes



```
poetry add pandas
```

## Realizar build do projeto



```
poetry build
```

# Estrutura de Pastas

- CookieCutter
- Jinja2

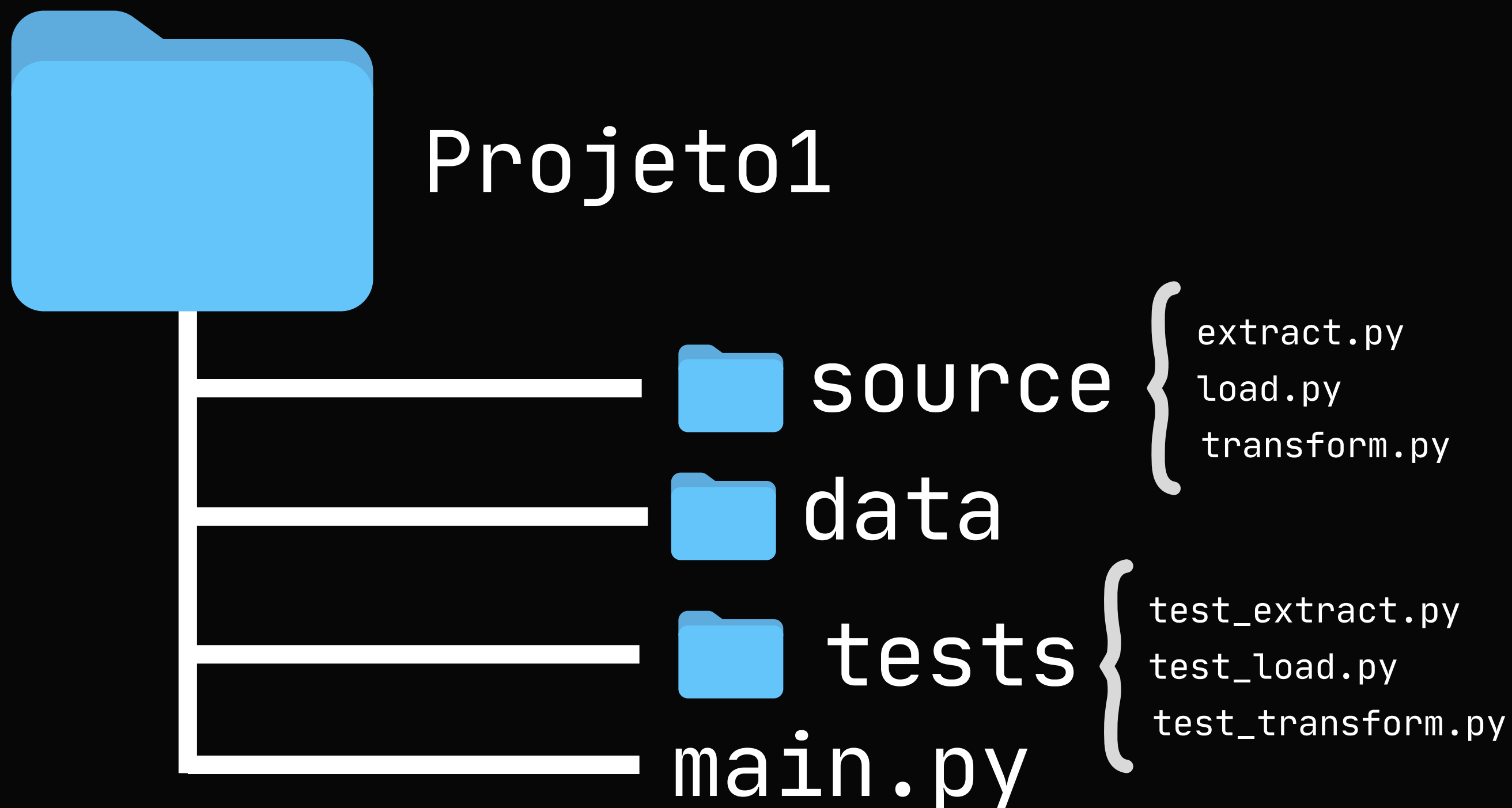


# O BOSS NÃO FOI DERROTADO AINDA...



## Projeto1

# O BOSS NÃO FOI DERROTADO AINDA...



# O BOSS NÃO FOI DERROTADO AINDA...



# LADOS NEGATIVOS DE NÃO SEGUIR PADRÃO DE PROJETO

- Dificuldade de **manutenção** de projetos simultâneos;
- Dificuldades de **onboarding** de novos membros nos projetos.



# Um item para ajudar nesse desafio...

COOKIECUTTER + jinja2



# CookieCutter

## Definição

**Linha de Comando que cria projetos a partir de modelos/templates**

- Funciona no Windows, Linux e MacOS;
- Vale para qualquer linguagem de programação e contexto de projeto;
- Utiliza a sintaxe do Jinja2.

# CookieCutter

## Instalação



```
pip install cookiecutter
```

## Execução usando pipx



```
pip install pipx
```

# CookieCutter

## Utilizar templates terceiros

```
pipx run cookiecutter <link> / <path>
```

- Cookiecutter Data Science

# Criando seu próprio template

- O projeto precisa estar no GitHub
- Dentro do repositório, deve ter o arquivo `cookiecutter.json`
  - Esse arquivo irá gerar o "formulário" para receber valores e armazenar em variáveis que podem ser usadas para customizar campos.
  - `"<variavel>": "valor padrão"`
  - `"<variavel>": ["escolha1", "escolha2"]`
  - Uso fora do arquivo: `{{cookiecutter.<variavel>}}`

# Conectando Tudo!

- Cookiecutter + Poetry

