

Gerenciando ambientes com Poetry

Izabelle Aller

Gerenciando dependências com Poetry

Izabelle Aller



London - 2018



- **Oceanógrafa**
- **Analista de dados junior na Dynamox**
- **Entusiasta por robótica**
- **Potterhead**
- **GitHub: @allerif**



DOCUMENTATION ▾

BLOG

HISTORY



PYTHON PACKAGING AND DEPENDENCY MANAGEMENT MADE EASY

Poetry



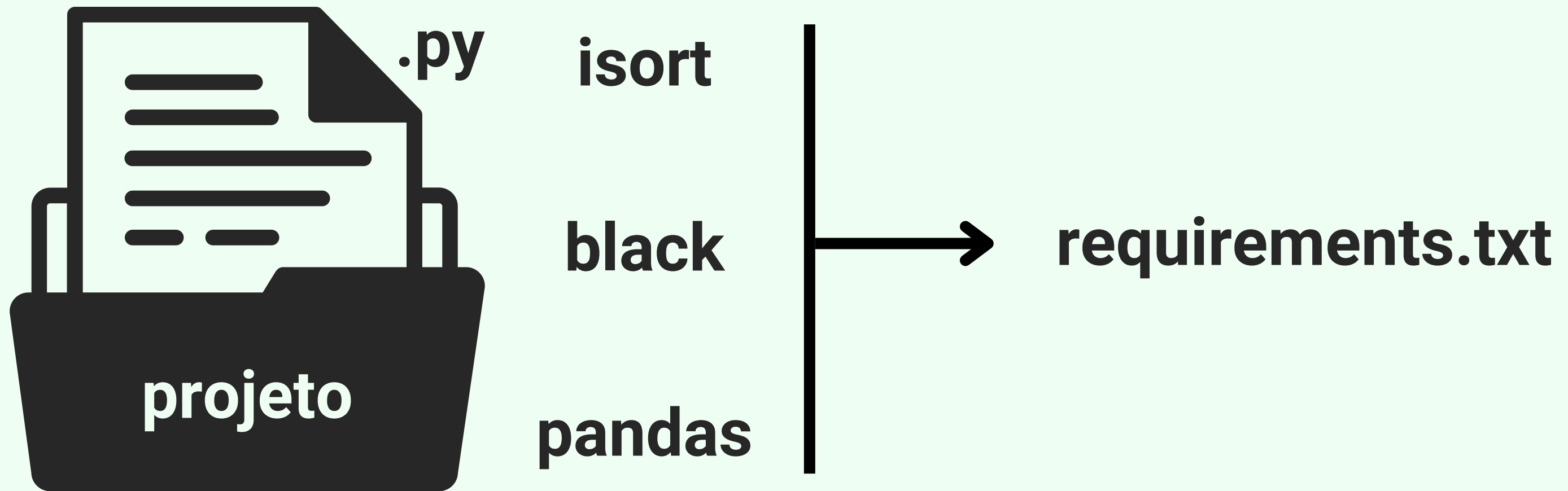
Por que usar um gerenciador?

- **Evitar conflitos de versão**
- **Manter estabilidade e consistência**
- **Aprimorar a segurança**
- **Simplificar a manutenção**
- **Facilitar a colaboração**

"Boas" práticas

em projetos python

pip install ...

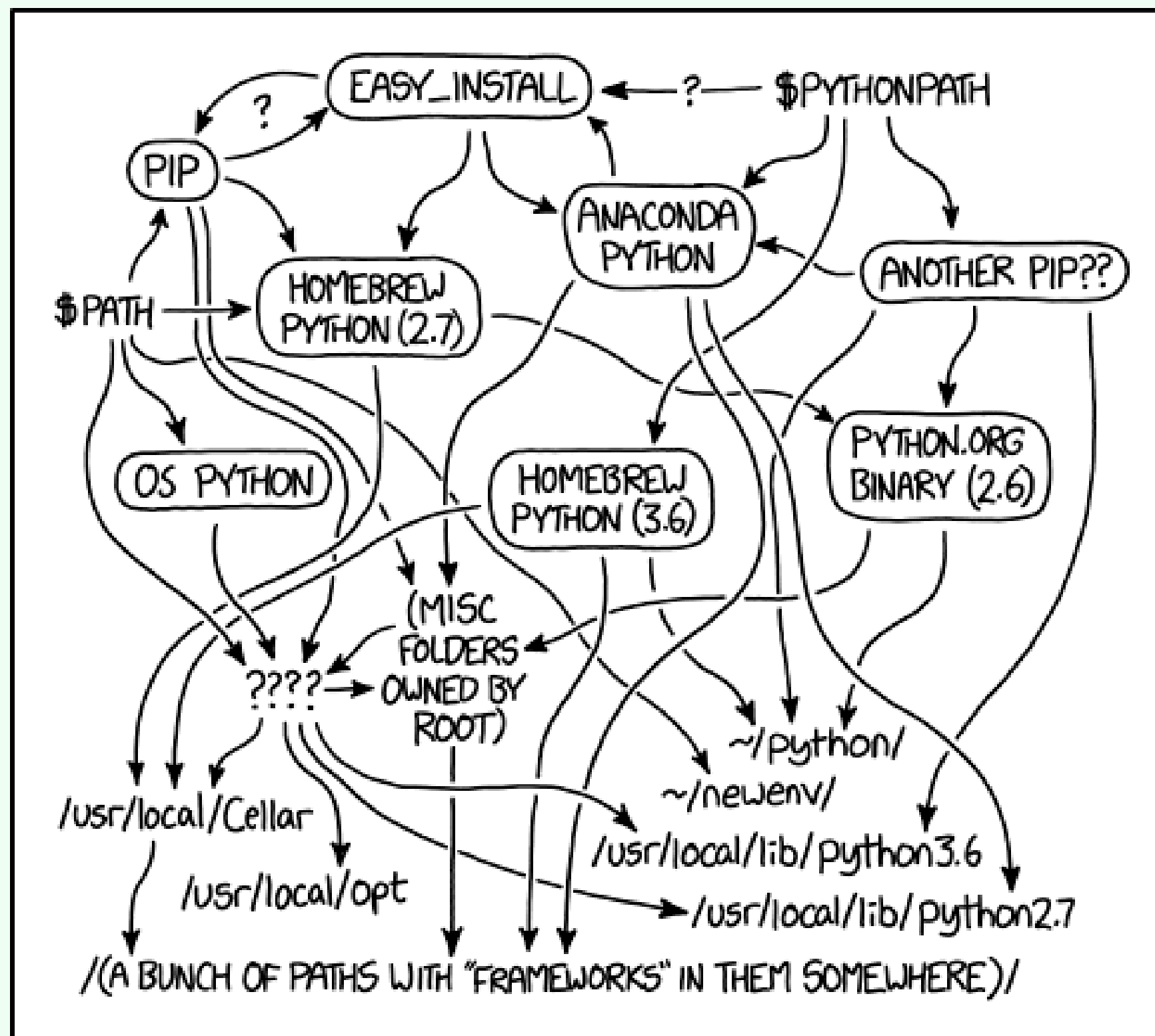




pandas 1.4

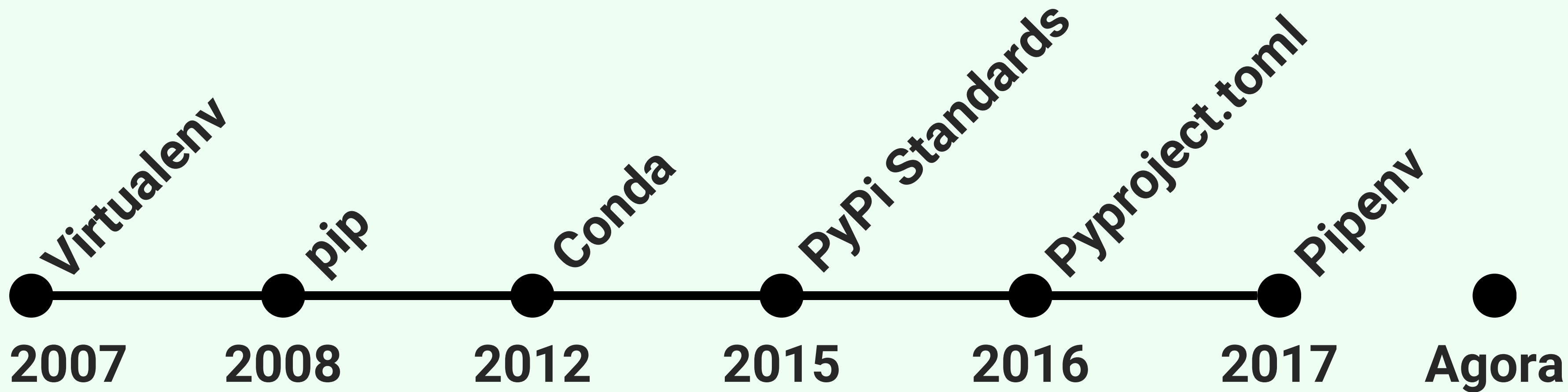


pandas 2.0



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Breve história



PEP 621

STORING PROJECT METADATA IN PYPROJECT.TOML

Pipenv

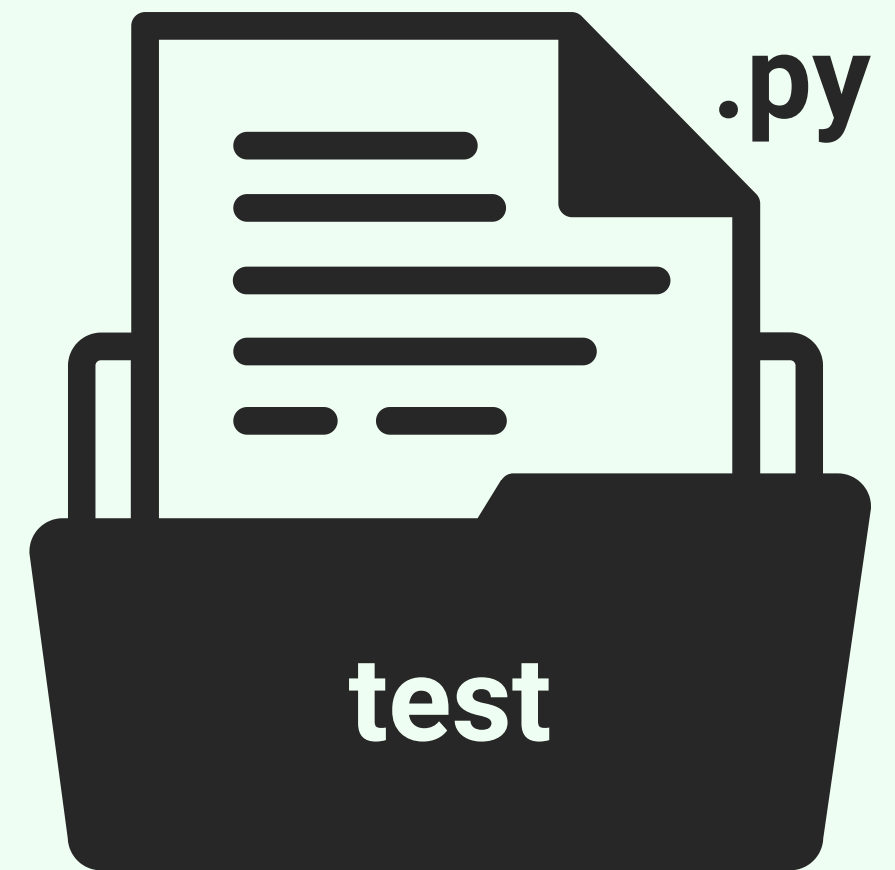
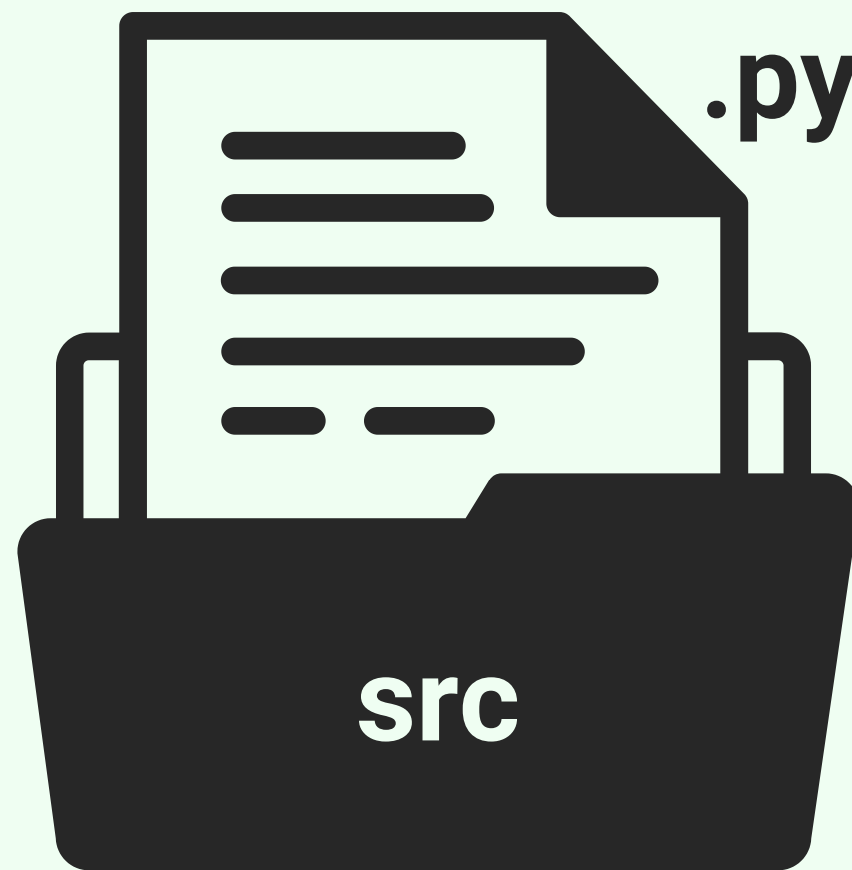
- **Pipfile**
 - **PyPi**
 - **Virtualenv + Pip**
-
- **Check**

Poetry

- **Pyproject.toml**
 - **PyPi**
 - **Virtualenv + Pip**
-
- **Distutils**
 - **Setuptools**
 - **pip-tools**
 - **PEP 440 + PyPi Standards**
 - **Builds independentes**

Comandos básicos

Workflow



Windows

```
(Invoke-WebRequest -Uri https://install.python-poetry.org -UseBasicParsing).Content | py -
```

```
--uninstall
```



```
poetry new .
```

```
poetry init
```

```
C:.
```

```
pyproject.toml
```


```
README.md
```

```
projeto
```

```
__init__.py
```

```
tests
```

```
__init__.py
```



```
[tool.poetry]
name = "projeto"
version = "0.1.0"
description = ""
authors = ["Allerif <izabelle.aller@gmail.com>"]
readme = "README.md"
```

```
[tool.poetry.dependencies]
python = "^3.10"
```

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

```
poetry shell
```

```
C:.\n  pyproject.toml\n  README.md\n  .venv\n  projeto\n    __init__.py\n  tests\n    __init__.py
```

```
poetry add --group docs mkdocs  
mkdocs new .
```

```
poetry remove --group docs mkdocs
```

```
C:\  
├── mkdocs.yml  
├── poetry.lock  
├── pyproject.toml  
├── README.md  
├── .venv  
├── docs  
│   └── index.md  
├── projeto  
│   ├── __init__.py  
│   └── tests  
│       └── __init__.py
```

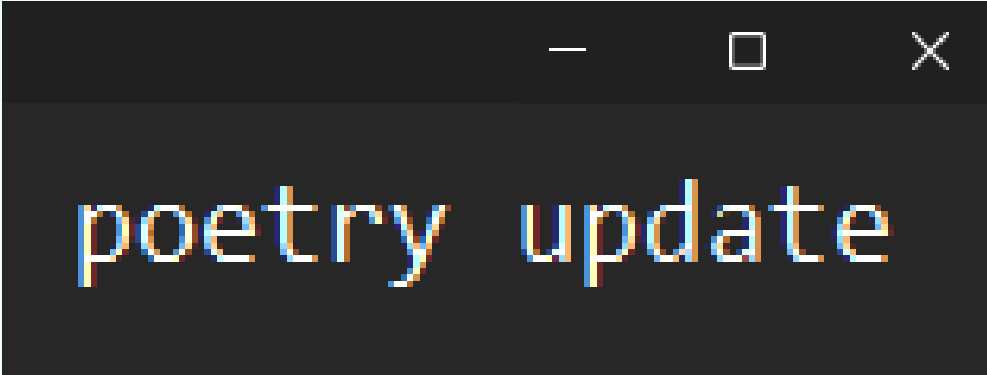
```
[tool.poetry]  
name = "projeto"  
version = "0.1.0"  
description = ""  
authors = ["Allerif <izabelle.aller@gmail.com>"]  
readme = "README.md"
```

```
[tool.poetry.dependencies]  
python = "^3.10"
```

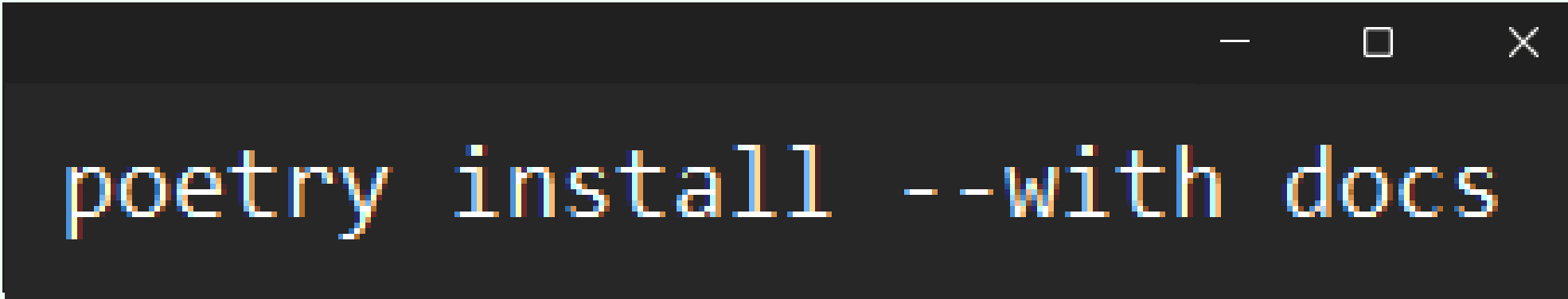
```
[tool.poetry.group.docs.dependencies]  
mkdocs = "^1.4.2"
```

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

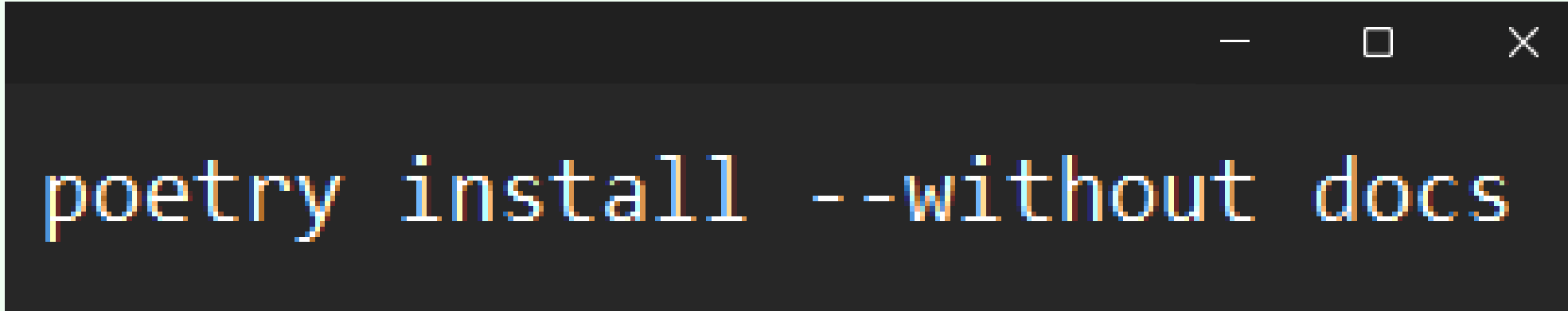
Outros comandos



```
poetry update
```

A terminal window with a dark background and a title bar containing standard window controls (minimize, maximize, close). The text 'poetry update' is displayed in a monospaced font.

```
poetry install --with docs
```

A terminal window with a dark background and a title bar containing standard window controls. The text 'poetry install --with docs' is displayed in a monospaced font.

```
poetry install --without docs
```

A terminal window with a dark background and a title bar containing standard window controls. The text 'poetry install --without docs' is displayed in a monospaced font.




Gerenciando pacotes e ambientes com Poetry - Live de Python #179



Share

PACOTES E AMBIENTES COM POETRY



Watch on  YouTube

Gerenciando dependências com Poetry

Izabelle Aller

Commit your poetry.lock file to version control

Committing this file to VC is important because it will cause anyone who sets up the project to use the exact same versions of the dependencies that you are using. Your CI server, production machines, other developers in your team, everything and everyone runs on the same dependencies, which mitigates the potential for bugs affecting only some parts of the deployments. Even if you develop alone, in six months when reinstalling the project you can feel confident the dependencies installed are still working even if your dependencies released many new versions since then. (See note below about using the update command.)



For libraries it is not necessary to commit the lock file.