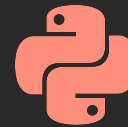




Ambientes virtuais e instalação de pacotes

Live de Python #191



1. Ambientes e bibliotecas

Um nivelamento básico

2. pip

Como instalar pacotes

3. requirements.txt

Como listar minhas dependências

4. venv

Como isolar os meus pacotes

5. Outras ferramentas

Complementado esse ecossistema

6. O futuro

O ambiente em constante evolução



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, Alysso 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, Fernando Silva, Flávio Meira, Flavkaze Flavkaze, Francisco Alencar, Franklin Silva, Gabriel Barbosa, Gabriel Sarmento, 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, 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, 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 Cunha, Natan Cervinski, Nicolas Teodosio, Nilton Pimentel, Osvaldo Neto, Patricia Minamizawa, Patrick Brito, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, Priscila Santos, Reinaldo Silva, Renan Moura, Richard Nixon, Riverfount Riverfount, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Junior, Rodrigo Vaccari, 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



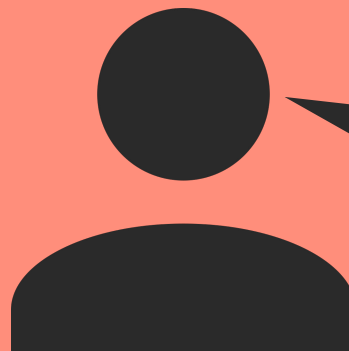
E suas bibliotecas

Ambientes

A ideia principal dessa live

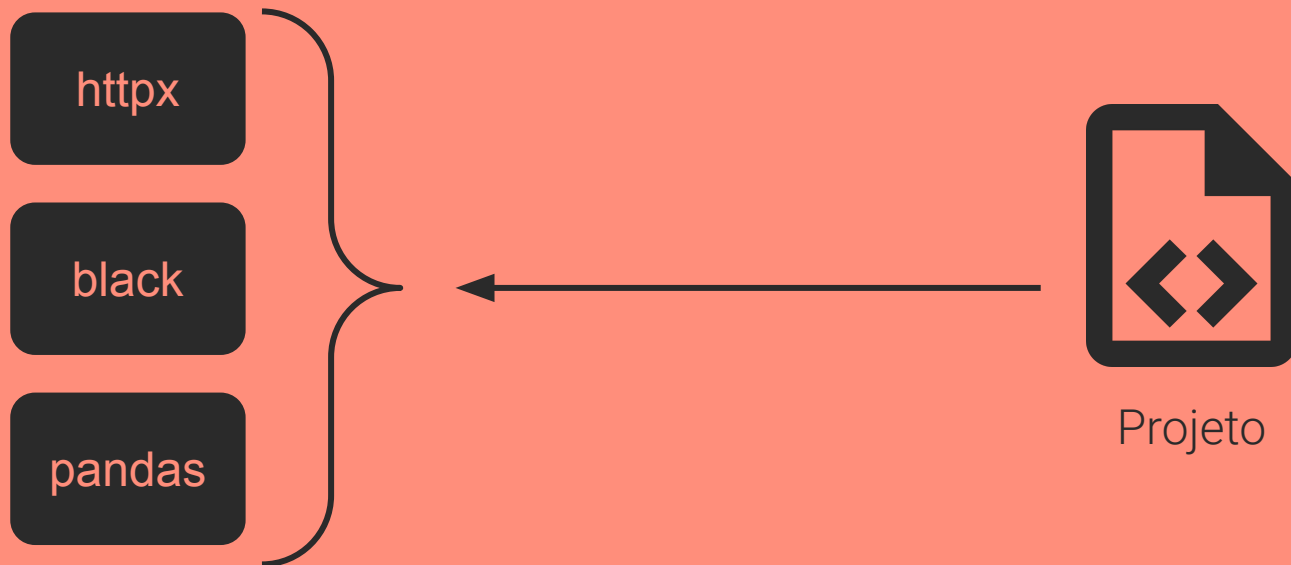


Projeto



Hmmm, preciso
de algumas
bibliotecas

A ideia principal dessa live



`pip install httpx black pandas`

A ideia principal dessa live



Pandas
1.4

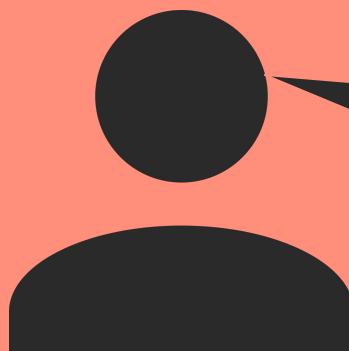


Projeto
A

Pandas
2.0



Projeto
B



pip install pandas?

A ideia principal dessa live



Pandas
1.4




Projeto A

Pandas
2.0

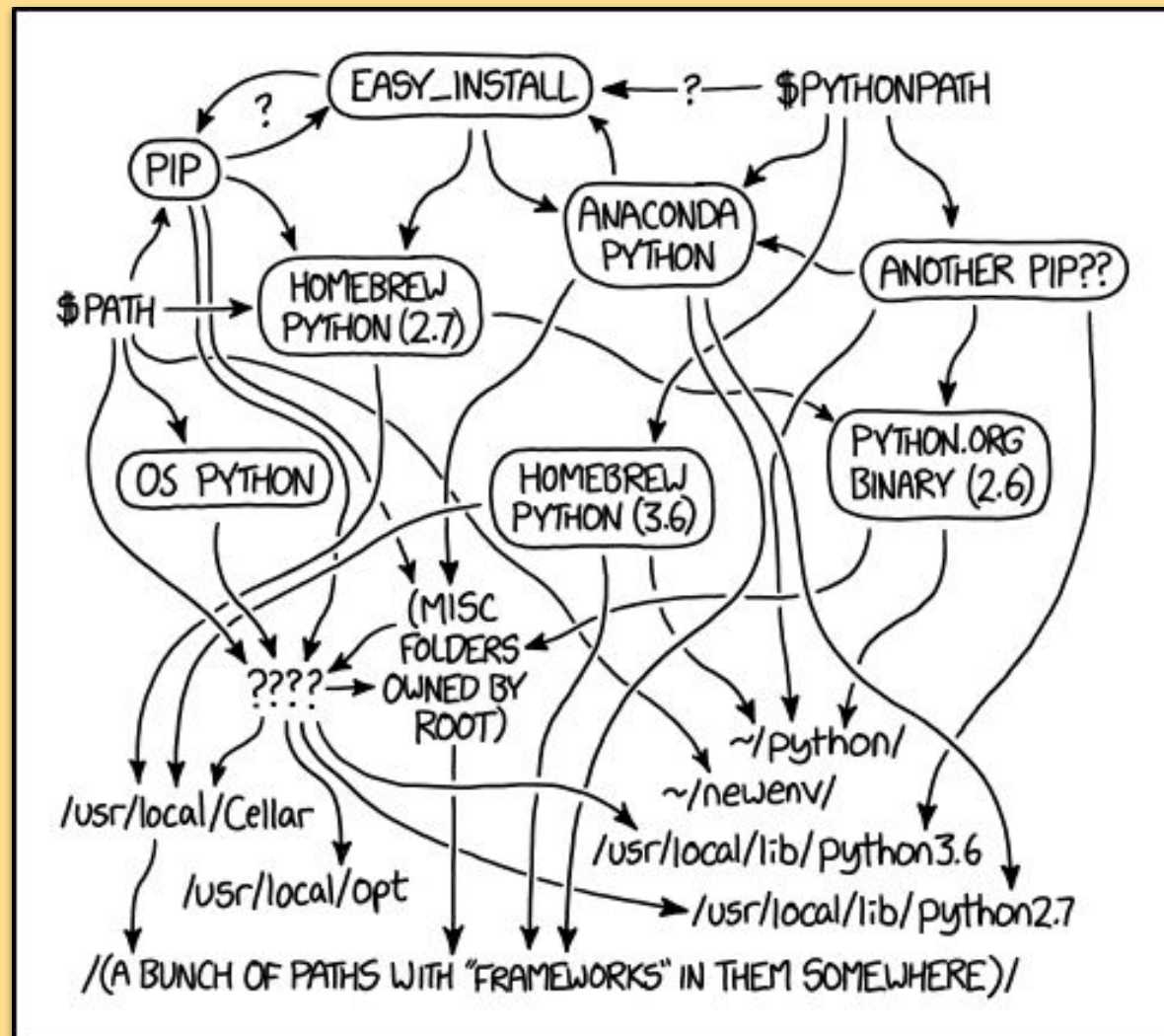


Projeto B





Já sei. Vou procurar na
internet como instalar
duas versões da
mesma biblioteca no
python



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

<https://xkcd.com/1987/>

GERENCIAMENTO DE BIBLIOTECAS NO PYTHON



**PIP
INSTALL**



**PYTHON
-M VENV**



VIRTUALENV



POETRY



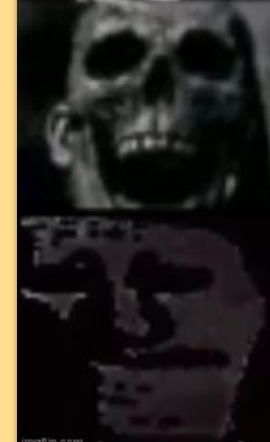
PIPX



VIRTUALENVWRAPPER



PIPENV



PDM

EASY_INSTALL

SITE.GETSITEPACKAGES()



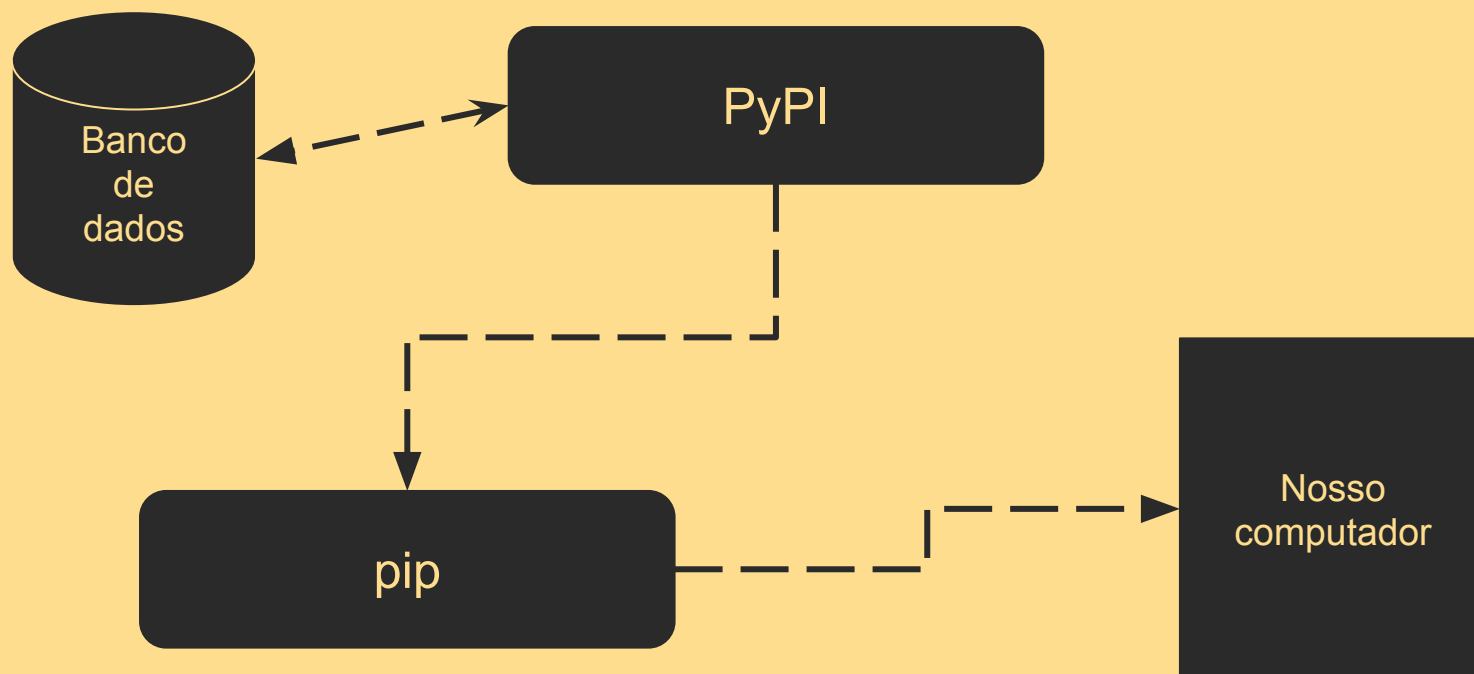
PIP

Começando do
começo

O que é o pip?



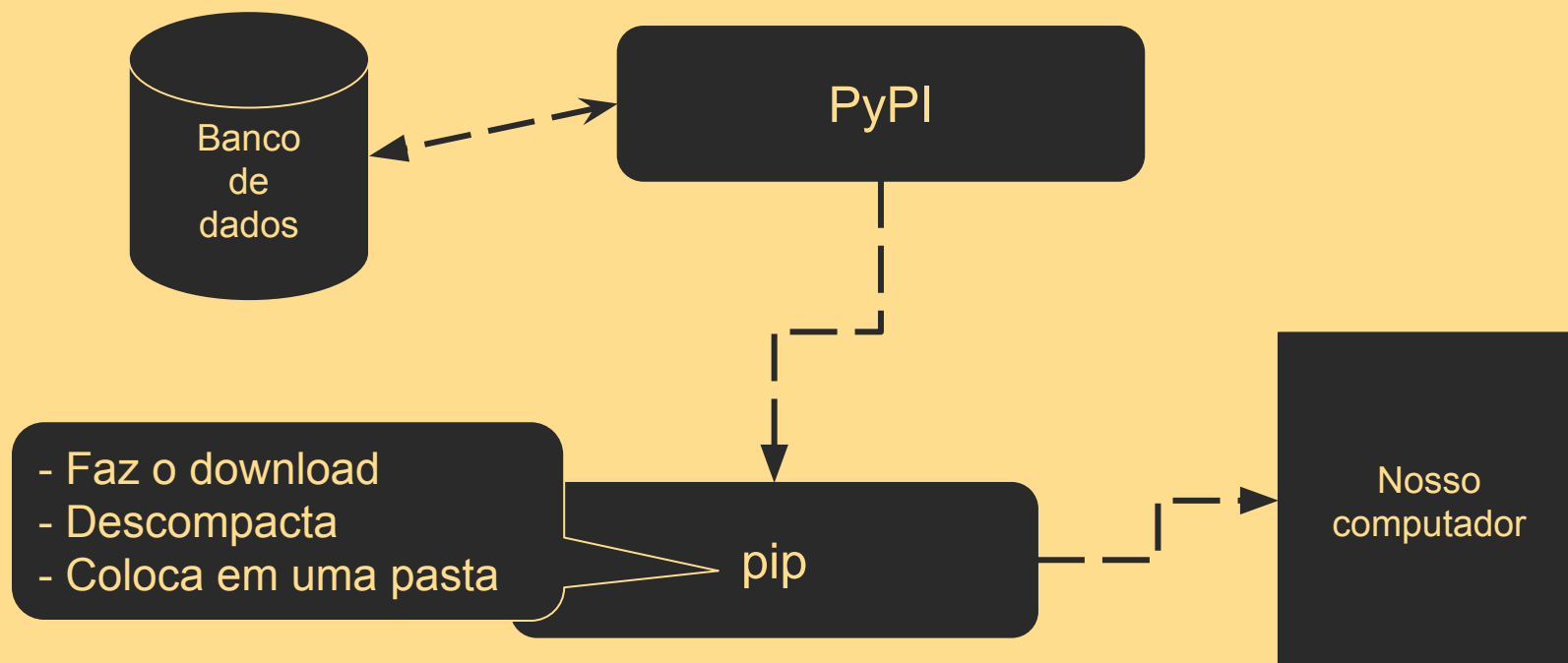
PIP - Pip Install Package. É uma maneira de instalar pacotes externos (3-party) no nosso ambiente.



O que é o pip?



PIP - Pip Install Package. É uma maneira de instalar pacotes externos (3-party) no nosso ambiente.



Onde fica essa pasta?



```
1  import site
2  site.getsitepackages()
3  # ['/usr/lib/python3.10/site-packages']
```


E como funciona o esquema de importação?



```
1  from httpx import get
```

E como funciona o esquema de importação?



```
1  from httpx import get
```

```
1  import sys
2  print(sys.path)
3  ['',
4   '/usr/lib/python3.10',
5   '/home/dunossauro/.local/lib/python3.10/site-packages',
6   '/usr/lib/python3.10/site-packages']
```

E como funciona o esquema de importação?



```
1 import sys
2 print(sys.path)
3 ['',
4   '/usr/lib/python3.10',
5   '/home/dunossauro/.local/lib/python3.10/site-packages',
6   '/usr/lib/python3.10/site-packages']
```



Uma olhada no esquema de diretórios



O site-packages
armazena uma pasta
com o pacote
descompactado.

Ele não sabe sobre
versões

```
1  # pip install httpx
2  ls <caminho>/site-packages
3  ├── anyio
4  ├── certifi
5  ├── charset_normalizer
6  ├── h11
7  ├── httpcore
8  ├── httpx
9  ├── idna
10 ├── pip
11 ├── pkg_resources
12 ├── rfc3986
13 ├── setuptools
14 └── sniffio
```

Caso tenha ficado a dúvida



```
1  $ pip show httpx
2  Name: httpx
3  Version: 0.22.0
4  Summary: The next generation HTTP client.
5  Home-page: https://github.com/encode/httpx
6  Author: Tom Christie
7  Author-email: tom@tomchristie.com
8  License: BSD
9  Location: /<caminho>/lib/python3.10/site-packages
10 Requires: certifi, charset-normalizer, httpcore, rfc3986, sniffio
```

Todas as dependências em árvore



```
# pip install pipdeptree
$ pipdeptree
httpx==0.22.0
- certifi [required: Any, installed: 2021.10.8]
- charset-normalizer [required: Any, installed: 2.0.11]
- httpcore [required: >=0.14.5,<0.15.0, installed: 0.14.7]
  - anyio [required: ==3.*, installed: 3.5.0]
    - idna [required: >=2.8, installed: 3.3]
    - sniffio [required: >=1.1, installed: 1.2.0]
  - certifi [required: Any, installed: 2021.10.8]
  - h11 [required: >=0.11,<0.13, installed: 0.12.0]
  - sniffio [required: ==1.*, installed: 1.2.0]
- rfc3986 [required: >=1.3,<2, installed: 1.5.0]
- sniffio [required: Any, installed: 1.2.0]
```

Os ambientes
virtuais

VENV

O que é um ambiente virtual?



Uma ferramenta para "hackear" o site-packages.

- Cria um ambiente "local"
- Isolado do "Global"
- Vem embutido no Python

Como criar um ambiente



No terminal

```
$ python -m venv <nome_do_seu_ambiente>
```

```
$ python -m venv venv # exemplo
```

Com isso, criamos um ambiente



Uma pasta com o nome que passamos é criada **venv**

```
py-3.10.2 babbage in ~/git/live_191/venv
$ ls -l venv # investigando a pasta venv
drwxr-xr-x - dunossauro 7 fev 08:45 bin
drwxr-xr-x - dunossauro 7 fev 07:41 include
drwxr-xr-x - dunossauro 7 fev 07:41 lib
lrwxrwxrwx 3 dunossauro 7 fev 07:41 lib64 -> lib
.rw-r--r-- 70 dunossauro 7 fev 07:41 pyvenv.cfg
```

Mostar no shell



Caso eu esqueça xD



Ativando o ambiente virtual



Plataforma	Shell	Comando para ativar
Unix	bash/zsh	\$ source <venv>/bin/activate
	fish	\$ source <venv>/bin/activate.fish
	csh/tcsh	\$ source <venv>/bin/activate.csh
	PowerShell Core	\$ <venv>/bin/Activate.ps1
Windows	cmd.exe	C:\> <venv>\Scripts\activate.bat
	PowerShell	PS C:\> <venv>\Scripts\Activate.ps1

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

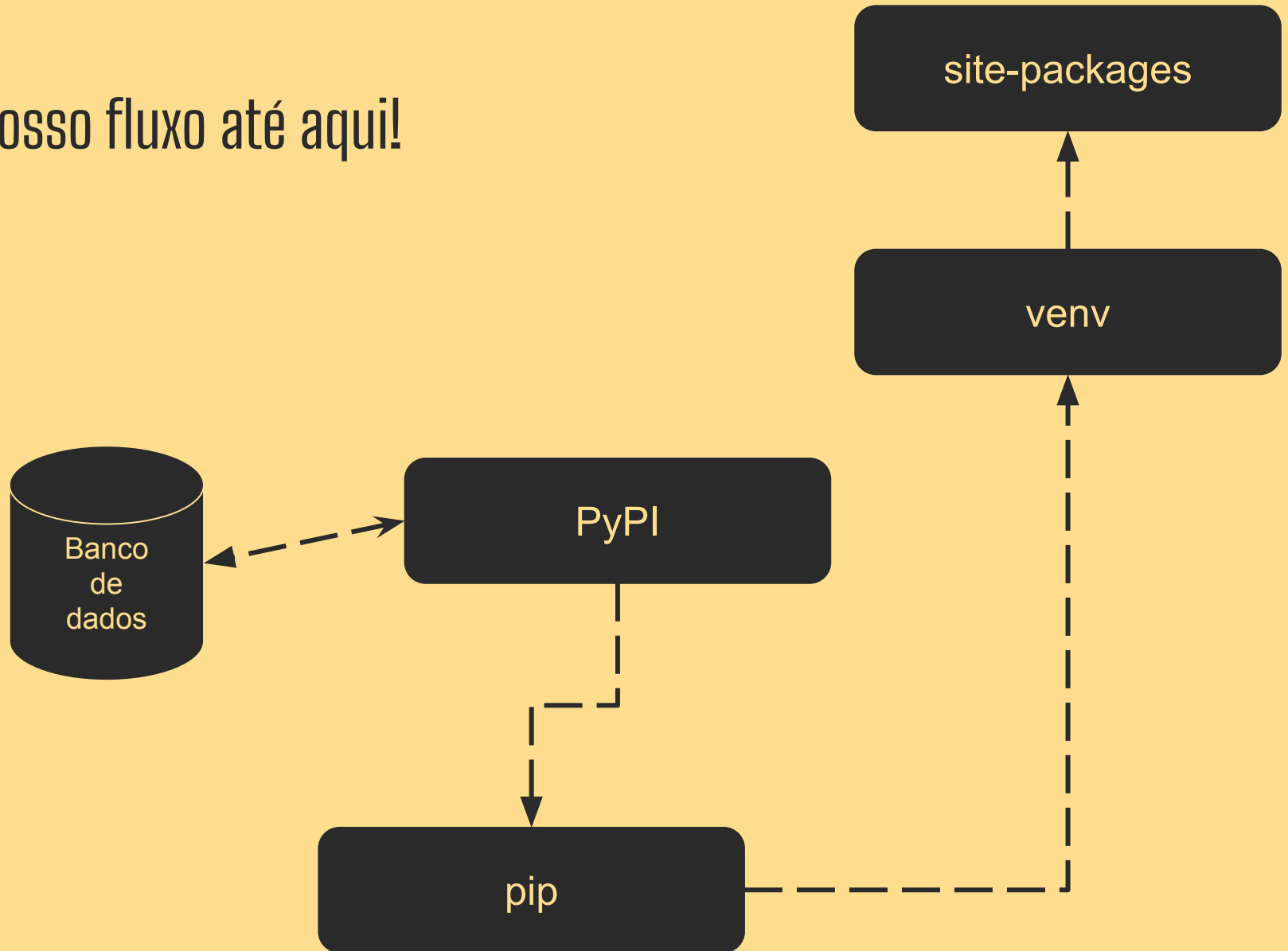
Ativando o ambiente virtual



Para desativar

```
$ deactivate
```

Nosso fluxo até aqui!



venv ou virtualenv?



Eles são e não são a mesma coisa




Qual a diferença?



☰ README.md

virtualenv

pypi **v20.13.1** implementation **cpython** | **pypy** python **2.7** | **3.5** | **3.6** | **3.7** | **3.8** | **3.9** | **3.10** docs **passing**

chat **55 online** downloads **24M/month** license **MIT**  check **passing** code style **black**

A tool for creating isolated **virtual** python environments.

- [Installation](#)
- [Documentation](#)
- [Changelog](#)
- [Issues](#)
- [PyPI](#)
- [Github](#)

Code of Conduct

Everyone interacting in the virtualenv project's codebases, issue trackers, chat rooms, and mailing lists is expected to follow the [PSF Code of Conduct](#).

<https://github.com/pypa/virtualenv>

Qual a diferença?



venv é uma **parte** da biblioteca **virtualenv**.

o venv foi embutido no python 3.4 e é somente uma parte de todo o virtualenv.

Vantagens do virtualenv

- É mais rápido
- Tem um ciclo de lançamento independente do python
- É extensível
- Tem um API de código

requ irem ents

Listando as
dependências

Listando minhas dependências



Imagine que você tenha que dizer a algum quais os pacotes necessários?

O arquivo **requirements.txt** é o lugar onde dizemos isso

Um exemplo



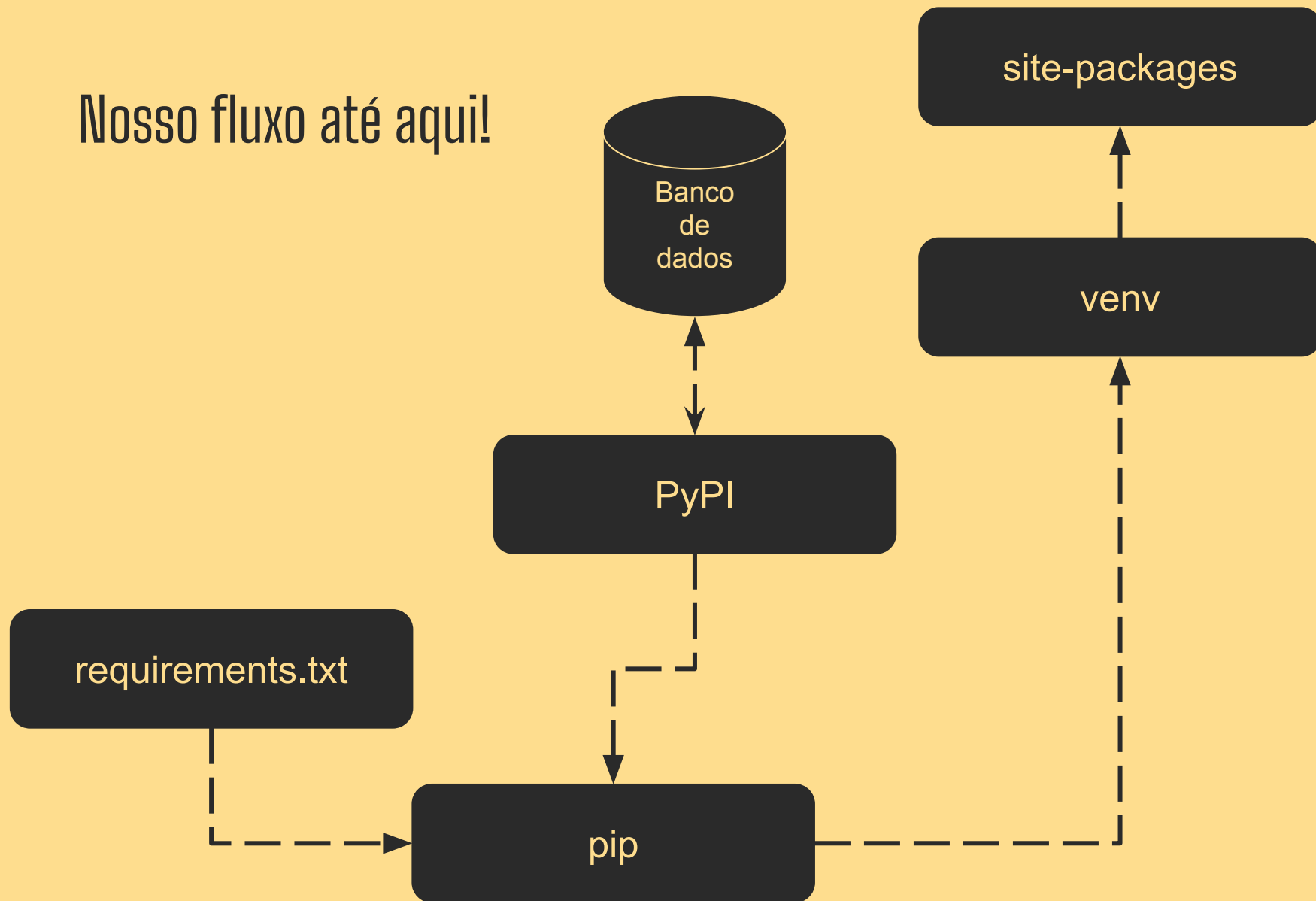
```
1  # requirements.txt
2
3  # Pacotes que quero instalar
4  httpx
5  pandas
6  pytest
7
8  # pacotes com versões específicas
9  black == 0.6.1    # igual a 0.6.1
10 django >= 4.1.1  # maior que 4.1.1
11 flask != 3.5     # diferente de 3.5
12 selenium ~= 1.1  # maior ou igual a 1.1, mas menor que 2
```

Como instalar os pacotes?



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

Nosso fluxo até aqui!



Um outro padrão



Já pensou quem nem todas as bibliotecas vão para o ambiente de produção? Pra isso, existe uma convenção de criar um arquivo chamado **requirements_dev.txt**

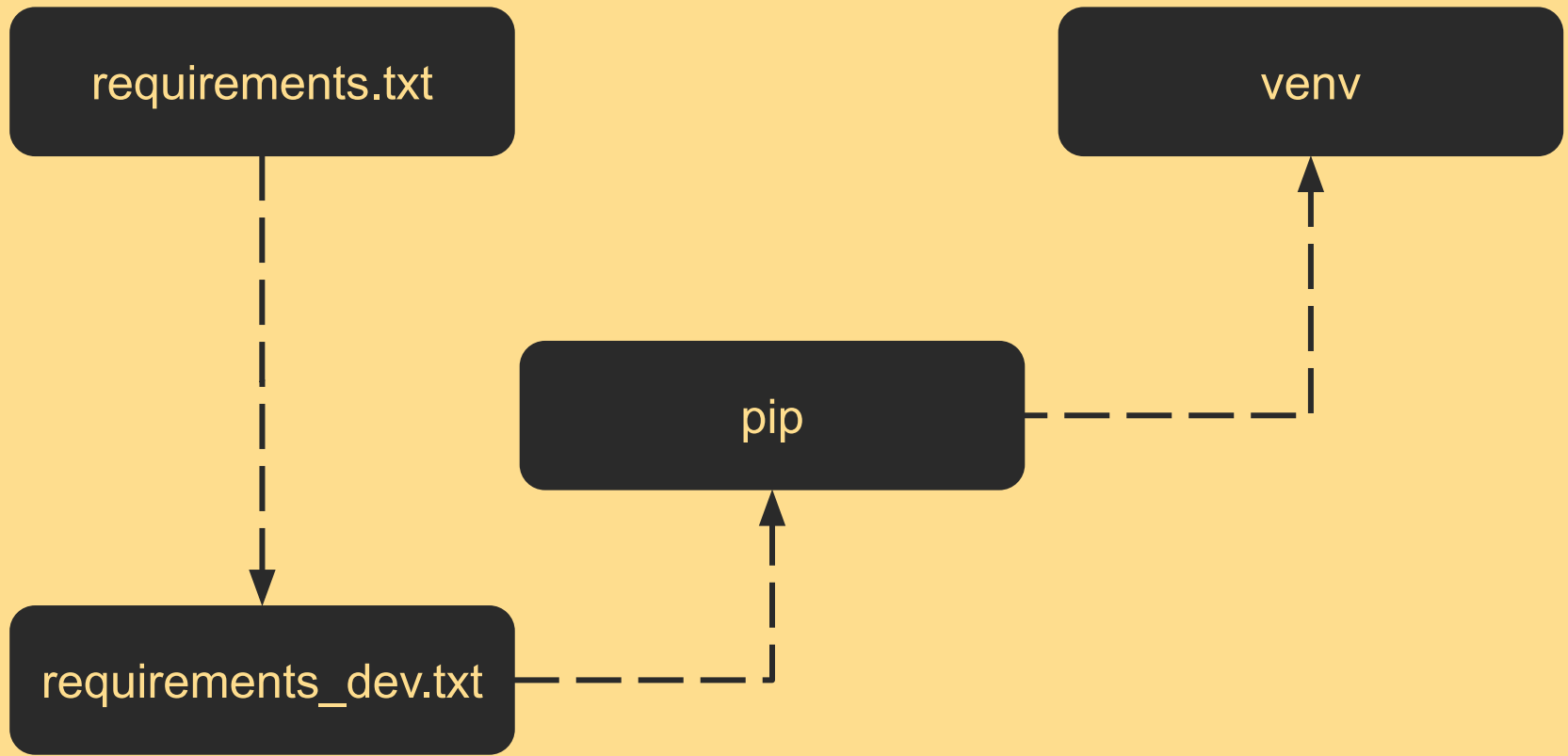
Nesse arquivo listamos as coisas de desenvolvimento e instalamos por ele, as de produção

Uma visão desse arquivo



```
1  # requirements_dev.txt
2  -r requirements.txt # instala as de prod
3
4  # bibliotecas de desenvolvimento
5  ipdb      # debugger
6  ipython  # shell interativo
7  black    # formatador de código
```


O que ficaria assim em desenvolvimento



Outro arquivo famoso



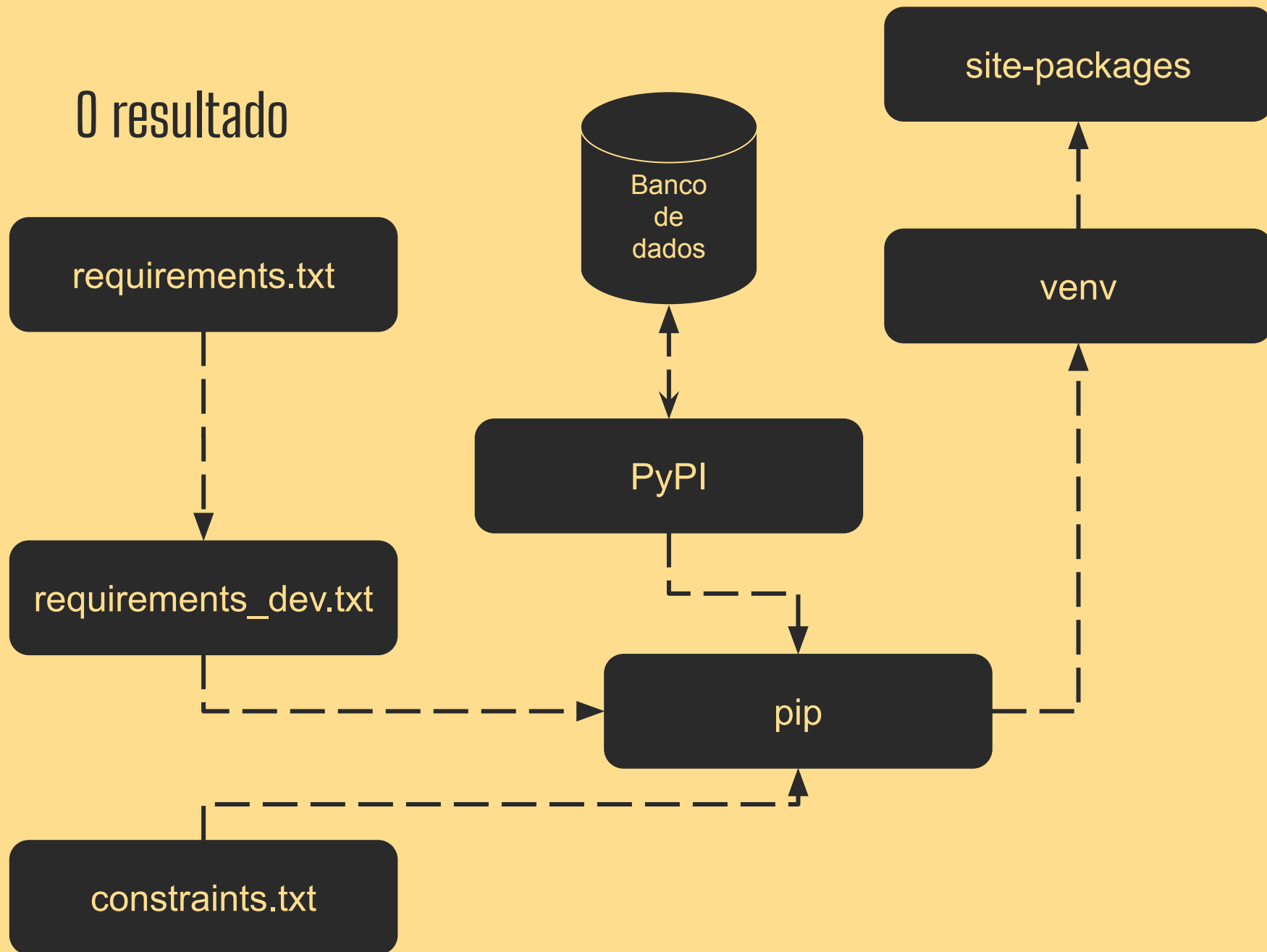
As vezes, por alguma limitação, decidimos que algumas bibliotecas **TEM** que estar em versões específicas. Por homologação, segurança, limitação do sistema e etc...

Para esses casos, existe o arquivo **constraints.txt**

Tem o mesmo formato do requirements.txt, mas não instala as bibliotecas, só força versões específicas

```
1  $ pip install -r requirements.txt -c constraints.txt
```

0 resultado



Outras
ferram
entas

Pq sim!

Só uma palinha sobre elas



Amigos do pip:

- **pip-autoremove**: Remove do ambiente as bibliotecas dependentes
- **pipdeptree**: Mostra quais bibliotecas depende de quais

Amigos do venv:

- **virtualenvwrapper**: Facilita a criação e manutenção dos ambientes

Amigos do Python:

- **pyenv**: Instala várias versões de python no ambiente
- **tox**: Roda testes em versões diferentes do python
- **pipx**: Instala ferramentas de linha de comando em um ambiente virtual isolado, **bom para instalações globais**

Velhos amigos



- **easy_install**: Antiga forma de instalar bibliotecas
- **pipenv**: Ferramenta usada para abstrair o venv e o pip juntos
- **Buildout**: Ferramenta para receitas de ambientes
- **Autoenv**: Ferramenta para gerenciar o ambiente fora do Python

O mundo científico



No ambiente de programação científica vamos nos deparar com outras ferramentas:

- **Conda**: Gerenciador de pacotes como o pip
- **miniconda**: Distribuição Python + Conda
- **Anaconda**: Distribuição do Python com muitos pacotes científicos
- **Mamba**: Parecido com o miniconda, mas refaz o conda em C++ para performance

0
futuro

pyproject.toml

O futuro das ferramentas



Estamos em fase de transição do empacotamento / gerenciamento de ambientes. O requirements.txt + requirements_dev.txt + setup.py serão substituídos pelo **pyproject.toml**

O futuro agora:

- **Poetry:**
 - Gerenciador de ambiente virtual (venv)
 - Gerenciador de empacotamento (setup.py + setup.cfg)
 - Metadados do pacote (setup.py)
 - Instalação de bibliotecas (pip)
 - Gerenciador de versões de bibliotecas (requirements.txt)
- **Flit:**
 - Quase o mesmo que o Poetry

O talvez futuro do ambiente virtual



Está sendo discutido, na PEP 582 o fim do virtualenv para gerenciamento local de pacotes. Sem que seja necessária a criação de ambientes virtuais.

Para experimentar esse futuro provável, você pode olhar o PDM



<https://pdm.fming.dev/>

Bonus



O meu ambiente

- **Pyenv:** Várias versões do python
- **pipx:** Instalação de ferramentas globais
 - blue
 - httpie
 - poetry
 - ipython
- **Poetry:** Gerenciamento de ambiente python



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3

