

□ Unidade 1 - Instalação de Bibliotecas no Python

Apresentação

Python é uma linguagem robusta de scripts que vem crescendo e se desenvolvendo no mundo inteiro, abarcando desde soluções simples até soluções de altíssima complexidade. Muito desse poder está na recursividade das bibliotecas Python, que são o grande diferencial para a expansão da linguagem, sendo esta considerada cada vez mais disruptiva para soluções robustas e aplicações simples.

Nesta Unidade de Aprendizagem, você vai aprofundar o seu conhecimento na rica linguagem Python, explorando o gerenciador de pacotes PIP, empregado para apoiar a orquestração e a gestão de bibliotecas na linguagem Python. Você também vai aprender a instalar as bibliotecas Python de forma conceitual e prática.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Reconhecer o aplicativo PIP.
- Analisar a necessidade de instalação de biblioteca.
- Praticar a instalação de bibliotecas.

Infográfico

Módulos e pacotes são conceitos básicos para se trabalhar com Python. Na criação de um pacote, além do programa ficar organizado, você pode importar os módulos de forma muito flexível.

Neste Infográfico, será apresentada, de forma esquematizada, a estrutura arquitetural de implementação de bibliotecas na linguagem Python, abarcando as funções do gerenciador PIP e as indexações para implementação e importação das bibliotecas.

{ DESCRIÇÃO ARQUITETURAL }

da implementação de bibliotecas Python

Python surgiu com o propósito de ser simples, acessível e fácil de usar. Nesse contexto, a estrutura arquitetural de implementação de bibliotecas na linguagem Python está dividida em:

BIBLIOTECAS EXTERNAS

Bibliotecas que necessitam de instalação para importação.

BIBLIOTECAS NATIVAS

Bibliotecas já disponíveis para importação.

INDEXAÇÃO NA LINGUAGEM PYTHON



APLICATIVO PIP

Com o PIP, é possível gerenciar pacotes de trabalho Python, ou seja, pode-se instalar, remover e atualizar pacotes em diversos projetos.



Os **pacotes** gerenciados pelo PIP são repositórios que contêm todos os arquivos necessários em um dado módulo do Python.

Todavia, os **módulos** são conjuntos de bibliotecas de código Python que podem ser incluídas nos projetos da linguagem, facilitando a implementação de programas.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Conteúdo do Livro

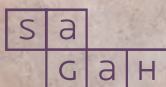
Python é uma linguagem que tem se difundido mundialmente dada a sua robustez e, ao mesmo tempo, a sua simplicidade de implementação e automação de *scripts*. Isso ocorre devido à grande rede de colaboradores que vêm desenvolvendo bibliotecas cada vez mais especializadas para a linguagem.

No capítulo Instalação de bibliotecas no Python, da obra *Programação Back End I*, você estudará a importância da implementação das bibliotecas dentro do contexto da linguagem Python. Além disso, analisará o agregador de pacotes PIP e, por fim, verá um exemplo da instalação de bibliotecas no Python.

Boa leitura.

PROGRAMAÇÃO BACK END I

Pedro Henrique Chagas Freitas



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Instalação de bibliotecas no Python

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Reconhecer o aplicativo pip.
- Analisar a necessidade de instalação de bibliotecas.
- Praticar a instalação de bibliotecas.

Introdução

Neste capítulo, abordaremos os principais conceitos sobre o aplicativo pip, além de analisarmos a necessidade de instalação de bibliotecas. Por fim, exemplificaremos a instalação de bibliotecas no Python.

Aplicativo pip

O aplicativo pip pode ser definido, de forma sucinta, como um gerenciador de pacotes Python. Segundo o site W3Schools, a função principal do pip é agregar pacotes ou módulos Python (PYTHON..., 2019, documento *on-line*). A partir da versão 3.4 de Python, o pip passou a ser incluído como padrão para o gerenciamento de pacotes. Os pacotes gerenciados pelo pip são repositórios que contêm todos os arquivos necessários em um módulo do Python. Já os módulos são bibliotecas de código Python que podem ser incluídas nos projetos da linguagem, facilitando a implementação de programas.

Para verificar se o pip está instalado no Python por padrão, basta executar em seu diretório de *scripts* a instrução a seguir.

```
C:\Users\[user]\AppData\Local\Programs\Python\Python36-32\Scripts>pip --version
```

**Link**

Caso o pip não seja encontrado, basta fazer o *download* e instalá-lo, acessando o *link* a seguir:

<https://qrgo.page.link/A35gA>

Por se tratar de um repositório, o pip também auxilia desenvolvedores Python a encontrar e instalar programas de *software* desenvolvidos na linguagem. Por este motivo, os desenvolvedores Python costumam utilizar o pip para distribuir o código-fonte de seus respectivos programas de *software* (TONSIG, 2008).

O pip auxilia no gerenciamento de pacotes de trabalho Python, permitindo instalar, remover e atualizar pacotes em diversos projetos.

**Link**

Há uma página de busca de pacotes disponíveis para *download* e instalação com o pip. Para procurar pacotes de bibliotecas, acesse o seguinte *link*:

<https://pypi.org/>

É possível pesquisar por um pacote específico por meio de palavras-chave, conforme mostra a Figura 1, com o exemplo da palavra **mysql**.

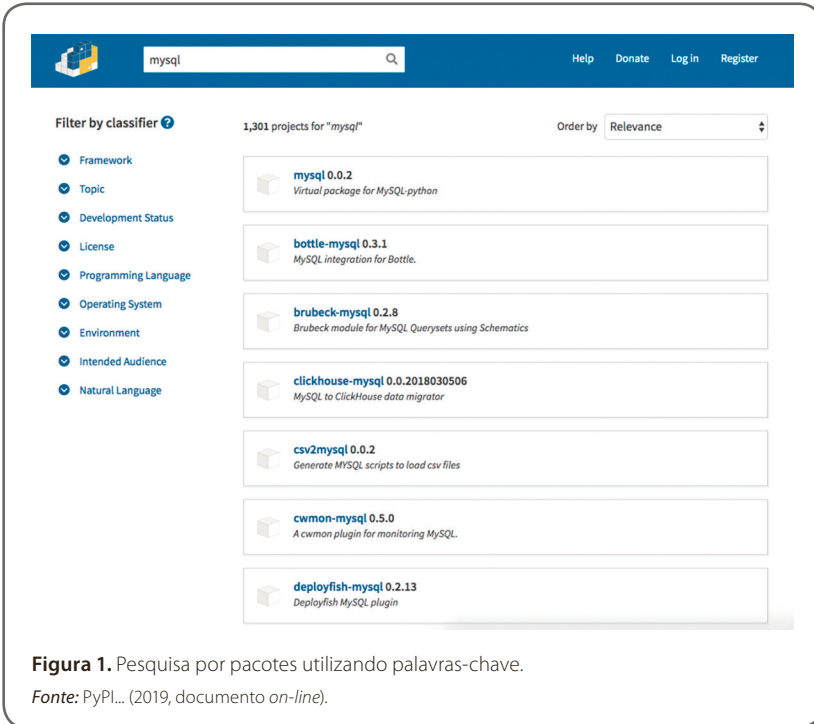


Figura 1. Pesquisa por pacotes utilizando palavras-chave.

Fonte: PyPI... (2019, documento on-line).

Para instalar o pip, baixe o arquivo `get-pip.py` e salve-o em um diretório python. Em seguida, execute o `get-pip.py`. Preferencialmente, este comando deve ser executado com privilégios de administrador. Se você estiver utilizando o Windows, localize o atalho do Prompt de Comandos no Menu Iniciar, clique com o botão direito do mouse sobre o ícone, e clique com o botão esquerdo em “Executar como administrador”. Se você estiver utilizando o Linux ou o macOS, digite o comando `sudo get-pip.py`. Caso ele já venha instalado, não é necessário seguir nenhuma destas instruções, uma vez que o pip já estará nativo na linguagem Python.

Após instalar o pip, você pode utilizá-lo em diversas tarefas, como já mencionadas: instalar, remover, listar e atualizar pacotes. Para a instalação de novos pacotes com pip, basta executar o comando: `pip install nome_do_pacote`.

Este comando baixará o pacote desejado e o instalará em nosso sistema operacional, como mostra a Figura 2. O nome do pacote pode ser encontrado na página oficial do pip.


```
~ → pip install mysqlclient
Collecting mysqlclient
  Using cached https://files.pythonhosted.org/packages/ec/fd/83329b9d3e14f7344d1cb31f128e6dbba70c5975c9e57896815dbb1988ad/my
Building wheels for collected packages: mysqlclient
  Running setup.py bdist_wheel for mysqlclient ... done
  Stored in directory: /Users/fagnerpsantos/Library/Caches/pip/wheels/71/9b/6f/a8c79bceb4a35f4acf49363883f958f31b113470a3b50
Successfully built mysqlclient
Installing collected packages: mysqlclient
Successfully installed mysqlclient-1.3.13
You are using pip version 10.0.1, however version 10.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

Figura 2. Baixando um pacote no sistema operacional.

Para listar os pacotes de instalação, utilizamos o comando `pip freeze`, conforme mostra a Figura 3.

```
~ → pip freeze
Django==1.11.14
mysqlclient==1.3.13
pytz==2018.5
virtualenv==16.0.0
```

Figura 3. Listando os pacotes de instalação.

Por fim, para realizar a instalação dos pacotes, utilizamos o comando: `pip install --upgrade nome_do_pacote`, conforme mostra a Figura 4.

```
~ → pip install --upgrade django
Collecting django
  Downloading https://files.pythonhosted.org/packages/f8/1c/31112c778b7a56ce18e3ff5e8915719f8e1cd3476c1eef557dddcfac8b/
ngo-1.11.15-py2.py3-none-any.whl (6.9MB)
100% |#####| 7.0MB 1.9MB/s
Requirement not upgraded as not directly required: pytz in /usr/local/lib/python2.7/site-packages (from django) (2018.5)
Installing collected packages: django
  Found existing installation: Django 1.11.14
  Uninstalling Django-1.11.14:
    Successfully uninstalled Django-1.11.14
  Successfully installed django-1.11.15
```

Figura 4. Instalação dos pacotes.

Dessa forma, ao utilizar o gerenciador `pip`, não será mais necessário baixar um pacote específico e instalá-lo manualmente, pelo seu arquivo de código-fonte. Ao contrário, por meio do `pip`, utilizamos apenas alguns comandos no terminal e implementamos a gerência dos pacotes.

Análise da necessidade de instalação de bibliotecas no Python

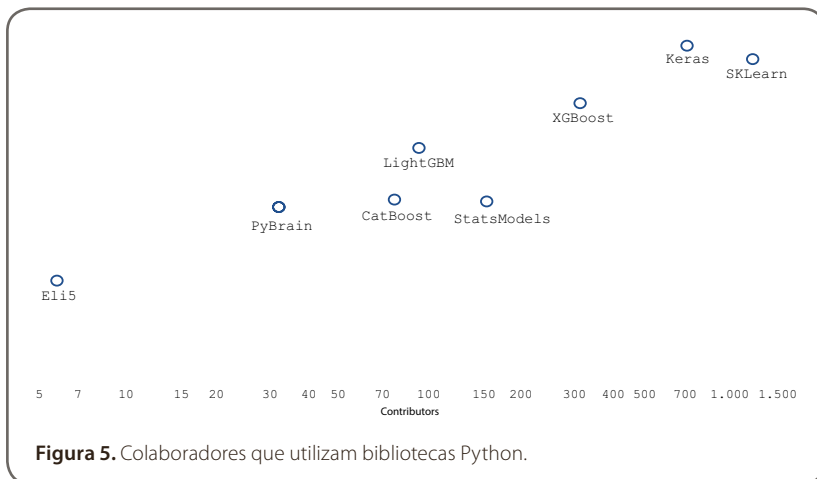
Existem diversos tipos de bibliotecas na linguagem Python, sendo que cada tipo destas está diretamente relacionada às funções necessárias para determinada implementação. Portanto, de acordo com o tipo de *software* e suas funcionalidades teremos bibliotecas específicas, diferenciando somente as bibliotecas nativas, que já estão internalizadas na linguagem, de outras bibliotecas que precisam ser importadas e instaladas, no caso, as bibliotecas externas (PRESSMAN, 2011).

Basicamente, uma biblioteca é um conjunto de novas instruções que podem ser utilizadas. Por exemplo, a biblioteca PyAudio é capaz de reproduzir áudio e outros elementos relacionados ao som, ou seja, ela foi criada para esta finalidade. Caso desejássemos desenvolver um *software* de reprodução de músicas em Python, provavelmente utilizaríamos esta biblioteca. Observe que não existe apenas uma biblioteca para dada finalidade, pois no geral, existem diversas bibliotecas para uma mesma finalidade (SOMMERVILLE, 2007).

A instalação de bibliotecas Python é importante pois demonstra a desacoplção da linguagem, isto é, não temos, necessariamente, todas as bibliotecas armazenadas na linguagem, mas podemos ter diversos tipos de bibliotecas sendo desenvolvidas neste exato momento, que podemos utilizar para a implementação de diversos cenários.

Isto demonstra o grande poder computacional por trás do Python. Por conseguinte, isto auxilia desenvolvedores do mundo inteiro a colaborar com a evolução desta linguagem, principalmente em cenários novos, como o *Machine Learning* (TONSIG, 2008), que tem apresentado diversos algoritmos criados em Python nos últimos anos, que utilizam bibliotecas como: *Eli5*, *PyBrain*, *CatBosst*, *LightGBM*, *StatsModels*, *XGBoostm*, *Keras* e *SKLearn*.

O gráfico apresentado na Figura 5 mostra a quantidade de colaboradores que atualmente utiliza estas bibliotecas.



A biblioteca SKLearn tem o maior número de colaboradores, com exatos 1.175 usuários, sendo responsável por realizar a análise de dados com *Machine Learning*. Já a biblioteca Keras implementa redes neurais para a experimentação e pesquisas.

O Python continua a liderar as pesquisas relacionadas a tarefas de *Machine Learning*, inteligência artificial (IA), *Deep Learning* e ciência de dados. De acordo com a builtin.com, 45% das empresas de tecnologia preferem utilizar o Python para implementar IA e *Machine Learning*, dado o alto nível de colaboração e riqueza de suas variações de bibliotecas.

Exemplo prático de instalação de biblioteca no Python

Nesta seção, aprenderemos a instalar uma biblioteca Python. Para título de exemplo, utilizaremos o ambiente do sistema operacional Windows, tendo em vista que este é o mais utilizado nos dias de hoje e nativo na maioria dos computadores pessoais.

Primeiramente, precisamos abrir o Prompt de Comando do Windows como administrador, como demonstra a Figura 6, com o seguinte comando:

```
net user Administrador /active:yes
```



```
CA Administrador: Prompt de Comando
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.
C:\Windows\system32>net user Administrador /active:yes
Comando concluído com êxito.

C:\Windows\system32>_
```

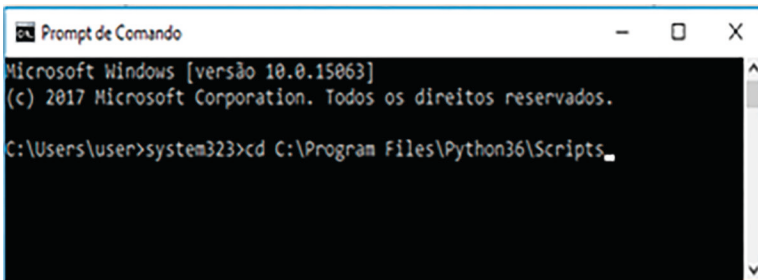
Figura 6. Executando o Prompt de Comando como administrador.

Em seguida, mudamos o diretório para a pasta `Scripts`, na pasta do Python.

No caso, estamos adotando a versão 3.6 instalada da linguagem Python. Logo, nosso comando será:

```
cd C:\Program Files\Python36\Scripts
```

A Figura 7 mostra a mudança para o diretório (pasta) desejado.



```
Prompt de Comando
Microsoft Windows [versão 10.0.15063]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.
C:\Users\user>system32>cd C:\Program Files\Python36\Scripts_
```

Figura 7. Como mudar para o diretório desejado utilizando o Prompt de Comando.

Após, basta utilizar o pip e atribuir o nome à biblioteca. No exemplo, vamos instalar a biblioteca PyAudio.

```
pip install pyaudio
```

Feito isso, a instalação é iniciada, como mostra a Figura 8.

```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32\>Cd C:\Program Files\Python36\Scripts

C:\Program Files\Python36\Scripts>pip install pyaudio
Collecting pyaudio
Using cached PyAudio-0.2.11-cp36-cp36m-win_and64.whl
Installing collected package: pyaudio
Successfully installed pyaudio-0.2.11

C:\Program Files\Python36\Scripts>
```

Figura 8. Iniciando a instalação da biblioteca

Com a biblioteca instalada, basta realizar o `import` (importar biblioteca) quando for desenvolver o código Python, por exemplo: `import pyaudio`.

Por fim, percebemos o alto poder computacional das possibilidades de utilização das bibliotecas Python, o que colabora com o grande crescimento de colaboradores que vem trabalhando na criação de outras bibliotecas para esta linguagem.



Referências

PRESSMAN, R. S. *Engenharia de software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH; Bookman, 2011. 780 p.

PYPI – The Python Package Index. *PyPI; Python Software Foundation*, Wilmington, 2019. Disponível em: <https://pypi.org/>. Acesso em: 11 nov. 2019.

PYTHON PIP. *W3Schools*, Sandnes, 2019. Disponível em: https://www.w3schools.com/python/python_pip.asp. Acesso em: 11 nov. 2019.

SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: Pearson Prentice Hall, 2008. 552 p.

TONSIG, S. L. *Engenharia de software: análise e projeto de sistemas*. 2. ed. Rio de Janeiro: Ciência Moderna, 2008. 319 p.



Fique atento

Os *links* para *sites* da Web fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integridade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Dica do Professor

Mundialmente difundida, a linguagem Python, criada em 1991, popularizou-se de uns anos para cá e tem colaboradores em diversos lugares do mundo, implementando bibliotecas especializadas e armazenando as suas funcionalidades em pacotes.

Nesta Dica do Professor, você verá, passo a passo, a criação de um pacote para recepcionar bibliotecas Python, além da execução dos pacotes com as bibliotecas inseridas.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) Python é uma linguagem de *script*. *Scripts* são programas escritos para um ambiente de execução que pode interpretar e automatizar a execução de tarefas. Os *scripts* Python podem ser implementados por meio da utilização de bibliotecas específicas. Nesse caso, a escolha dessas bibliotecas depende, necessariamente:
 - A) da classe e dos objetos que compõem as bibliotecas.
 - B) do tipo de aplicação PIP.
 - C) do tipo e da característica da funcionalidade que se deseja implementar.
 - D) do tipo de sistema operacional e das classes que compõem as bibliotecas.
 - E) do tipo de aplicação PIP e dos métodos que compõem as bibliotecas.

- 2) O Python pode ser utilizado para construir qualquer sistema, desde coisas básicas até aplicações extremamente complexas. Para facilitar a utilização e a implementação dos *scripts* Python, pode-se empregar o PIP, que é um:
 - A) agregador e orquestrador de pacotes ou módulos do Python.
 - B) armazenador e disseminador de classes especializadas do Python.
 - C) armazenador e disseminador de métodos especializados do Python.
 - D) agregador e orquestrador de classes e objetos do Python.
 - E) agregador e orquestrador de funcionalidades nativas do Python.

- 3) O Python foi pensado para apresentar a sua estrutura com bibliotecas nativas e bibliotecas externas, que podem ser instaladas posteriormente. Dentro desse contexto, é comum utilizar o PIP para instalar, remover e atualizar pacotes que contêm bibliotecas Python. Qual o *script* utilizado para instalar o gerenciador PIP?
 - A) set-pip.py
 - B) get-pip.py

- C) pip.py
- D) execute.pip-py
- E) py-start.pip

4) Depois de instalar o gerenciador PIP, é possível utilizá-lo para várias tarefas, como, por exemplo, instalar, remover, listar e atualizar pacotes. Pensando na instalação de novos pacotes contendo bibliotecas Python com PIP, nesse caso, basta executar o comando:

- A) install nome_do_pacote
- B) pip.py install nome_do_pacote
- C) pip.set install nome_do_pacote
- D) pip.get.py install nome_do_pacote
- E) pip install nome_do_pacote

5) A linguagem Python pode ser utilizada em diversos sistemas operacionais. No ambiente do sistema operacional Windows, por exemplo, para realizar a instalação de uma biblioteca Python, basta utilizar, no *prompt* de comando, o comando:

- A) pip install nome_da_biblioteca. Em seguida, basta executar no código a instrução assign nome_da_biblioteca.
- B) install nome_da_biblioteca. Em seguida, basta executar no código a instrução import nome_da_biblioteca.
- C) pip install nome_da_biblioteca. Em seguida, basta executar no código a instrução import nome_da_biblioteca.
- D) pip install.py nome_da_biblioteca. Em seguida, basta executar no código a instrução set nome_da_biblioteca.
- E) pip.py install nome_da_biblioteca. Em seguida, basta executar no código a instrução get nome_da_biblioteca.

Na prática

Muitos desenvolvedores têm trabalhado no desenvolvimento de bibliotecas Python para *Data Science* (Ciência de Dados). Dentre estas, três se destacam, pois são empregadas em larga escala e têm ajudado na criação de outras bibliotecas mais especializadas.

Conheça a seguir, Na Prática, essas bibliotecas de destaque e as suas aplicações.

{ BIBLIOTECAS PYTHON PARA *DATA SCIENCE* }

Amplamente utilizada por ser flexível e de código aberto, a linguagem Python oferece diversos tipos de bibliotecas utilizados para manipulação de dados. Cada tipo está ligado diretamente às funções necessárias para determinada implementação. **A instalação de bibliotecas Python é importante, pois demonstra a desacoplação da linguagem.**



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

BIBLIOTECAS PRINCIPAIS

NumPy

O NumPy é destinado ao processamento de grandes matrizes e matrizes multidimensionais, e uma extensa coleção de funções matemáticas de alto nível e métodos implementados possibilitam a execução de várias operações com esses objetos.



Nos últimos anos, um grande número de melhorias foi feito na biblioteca. Além das correções de *bugs* e problemas de compatibilidade, as mudanças cruciais dizem respeito às possibilidades de estilo, ou seja, o formato de impressão dos objetos NumPy.

SciPy

Outra biblioteca central para computação científica é o SciPy. Ele é baseado no NumPy e, portanto, estende seus recursos. A principal estrutura de dados SciPy é novamente uma matriz multidimensional, implementada pelo NumPy. O pacote contém ferramentas que ajudam a resolver álgebra linear, teoria da probabilidade, cálculo integral e muitas outras tarefas.



A biblioteca recebeu grandes melhorias na construção, na forma de integração contínua em diferentes sistemas operacionais, novas funções e métodos, e o que é especialmente importante – os otimizadores atualizados.

Pandas

Pandas é uma biblioteca Python que fornece estruturas de dados de alto nível e uma grande variedade de ferramentas para análise. A grande característica desse pacote é a capacidade de traduzir operações muito complexas com dados em um ou dois comandos. O Pandas contém muitos métodos internos para agrupar, filtrar e combinar dados, bem como a funcionalidade de séries temporais. Tudo isso é seguido por indicadores de velocidade impressionantes.



A importância dessas bibliotecas está no fato de que elas são utilizadas em quase todas as soluções de *Data Science* desenvolvidas em Python e deram origem para outras bibliotecas de análise de dados, mineração de bases de dados e, inclusive, de inteligência artificial.