Pandas Python: o que é, para que serve e como instalar

Marcus Almeida
Atualizado em 16 de Agosto

Introdução

A Ciência de Dados é um ramo que vem ganhando cada vez mais notoriedade, várias empresas de pequeno a grande porte, como a Netflix, Airbnb e Google já possuem atividades de tomada de decisão baseadas em dados. Nesse cenário, a linguagem Python é bastante utilizada devido a sua versatilidade e simplicidade, contando com uma vasta quantidade de bibliotecas, e entre elas, o **Pandas**, uma das ferramentas essenciais quando se fala em Ciência de Dados.

Neste artigo, vamos conhecer a biblioteca Pandas, entender sobre as suas estruturas básicas, e também como instalar a ferramenta.

Confira neste artigo:

- Introdução
- O que é Pandas?
- Como funciona o Pandas?
- Como o Pandas é utilizado?
- Instalação do Pandas
- Conclusão

Mas o que exatamente é Pandas?

O que é Pandas?

Pandas é uma biblioteca para Ciência de Dados de código aberto (*open source*), construída sobre a linguagem Python, e que providencia uma abordagem **rápida** e **flexível**, com estruturas robustas para se trabalhar com dados relacionais (ou rotulados), e tudo isso de maneira simples e intuitiva.

Apesar do nome da biblioteca ser associado ao mamífero da família de ursos, tal qual o Python é associado com a espécie de cobra erroneamente, o nome da biblioteca Pandas é derivado do termo **Pan**el **Da**ta, um conceito em inglês relacionado ao campo de estudo da econometria.

De maneira geral, o Pandas pode ser utilizado para várias atividades e processos, entre eles: **limpeza e tratamento de dados**, **análise exploratória de dados (EDA)**, suporte em atividades de Machine Learning, consultas e queries em bancos de dados relacionais, visualização de dados, *webscraping* e muito mais. E além disso, também possui ótima integração com várias

outras bibliotecas muito utilizadas em Ciência de Dados, tais como: Numpy, Scikit-Learn, Seaborn, Altair, Matplotlib, Plotly, Scipy e outros.

Como funciona o Pandas?

Dentro do pacote Pandas, temos dois objetos primários importantes: as ***Series*** e os ***DataFrames***. E para entender um pouco melhor sobre essas estruturas, vamos utilizar como exemplo um conjunto de dados chamado *Iris*, que traz algumas informações a respeito de características de espécies das flores de Íris.

Series

As *Series* são objetos de tipo array **unidimensional**, com um eixo de rótulos, também chamado de *index*, que é responsável por identificar cada registro. Um exemplo de Series no Pandas é encontrado no dataset Iris quando isolamos uma das variáveis para exibição, por exemplo o comprimento da pétala (`PetalLengthCm`), onde podemos observar o seguinte formato:

```
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: PetalLengthCm, Length: 150, dtype: float64
```

A coluna de números antes dos espaços à esquerda é o index, e os dados são apresentados à direita. No final da apresentação, há uma pequena descrição de nome, formato e tipo de dados presentes na Series.

DataFrame

Os *DataFrames* são objetos **bidimensionais**, de tamanho variável. O seu formato é de uma **tabela**, onde os dados são organizados em linhas e colunas. Além disso, enquanto podemos pensar a Series como uma única coluna, o DataFrame seria uma união de várias Series sob um mesmo index. A estrutura do DataFrame é apresentada na seguinte imagem:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

Nós podemos trabalhar com a criação de cada uma dessas estruturas usando os métodos do Pandas (`pandas.DataFrame` e `pandas.Series`) sobre estruturas nativas do Python (como listas, arrays e dicionários). Também podemos trabalhar com a leitura e escrita de vários tipos de arquivos de dados, tais como:

- CSV;
- Planilhas do Excel;
- Parquet;
- SQL;
- HTML;
- JSON;
- XML;
- e muito mais.

Pandas e o Excel

Devido a adesão do mercado ao pacote Office da Microsoft, que oferece o editor de planilhas Excel, surgem discussões de porque utilizar o Pandas. Existem diferenças na proposta de cada software. Além do Pandas ser uma solução de código aberto e não proprietária, ao contrário do Excel, também é possível observar diferenças na quantidade de informação que cada um pode portar.

No Excel, o limite de construção das tabelas é de 1.048.576 linhas por 16.384 colunas. Já no Pandas, a limitação é baseada na quantidade de memória disponível, então podemos ter uma grande variedade de linhas e colunas desde que a memória alocada não ultrapasse a quantidade disponível na sua máquina.

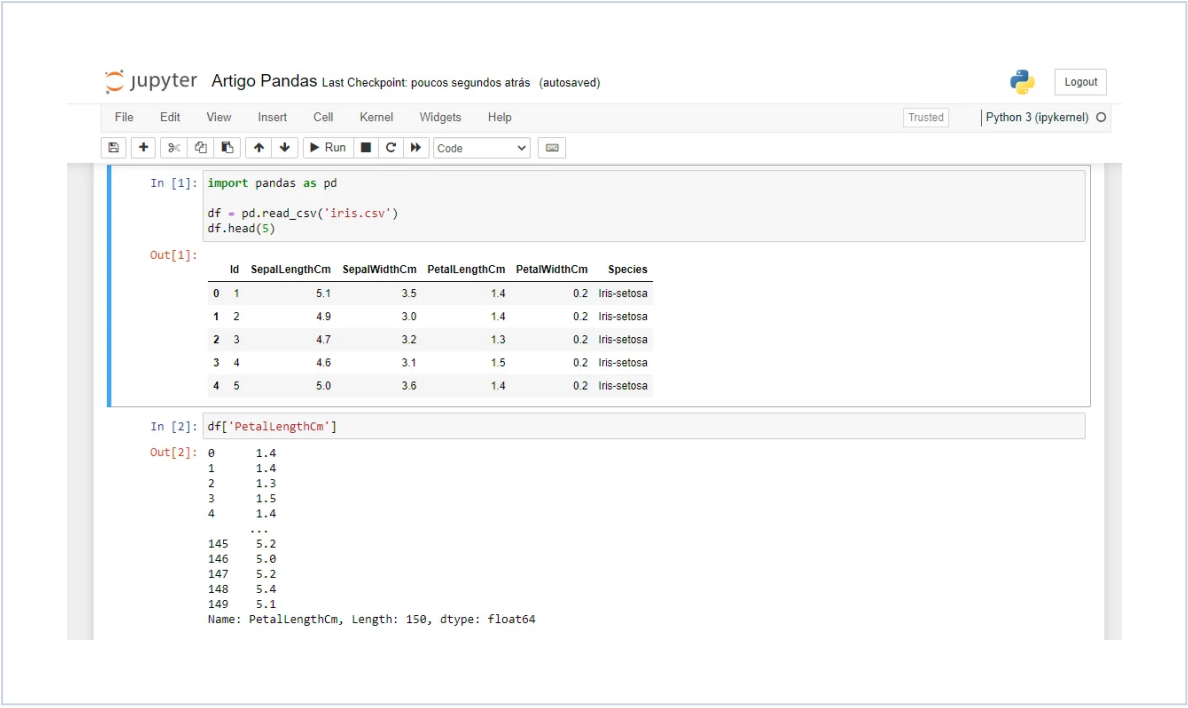
Conhecer os limites de cada ferramenta passa a ser interessante quando surge a necessidade de trabalhar com ambientes com maior quantidade de dados, e até mesmo em casos extremos, que ultrapassam facilmente os milhões de registros, como o cenário de *Big Data*.

Mas, ao mesmo tempo que esses softwares apresentam propostas diferentes, eles também podem ser trabalhados de maneira conjunta, já que o Pandas oferece **compatibilidade** com os arquivos do Excel, tanto em criação, em leitura, como também em escrita.

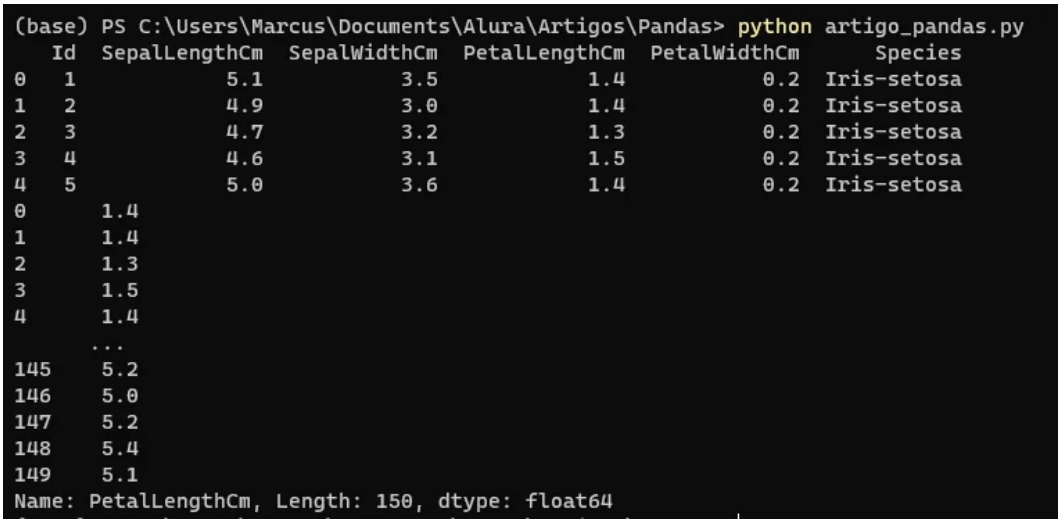
Como o Pandas é utilizado?

No dia a dia de um cientista de Dados, o Pandas é bastante utilizado em conjunto a notebooks interativos Python (arquivos com extensão .ipynb), tais como o Jupyter Notebook, no qual o Google Colab também é baseado. A ideia principal é aproveitar a boa apresentação do código e as suas saídas,

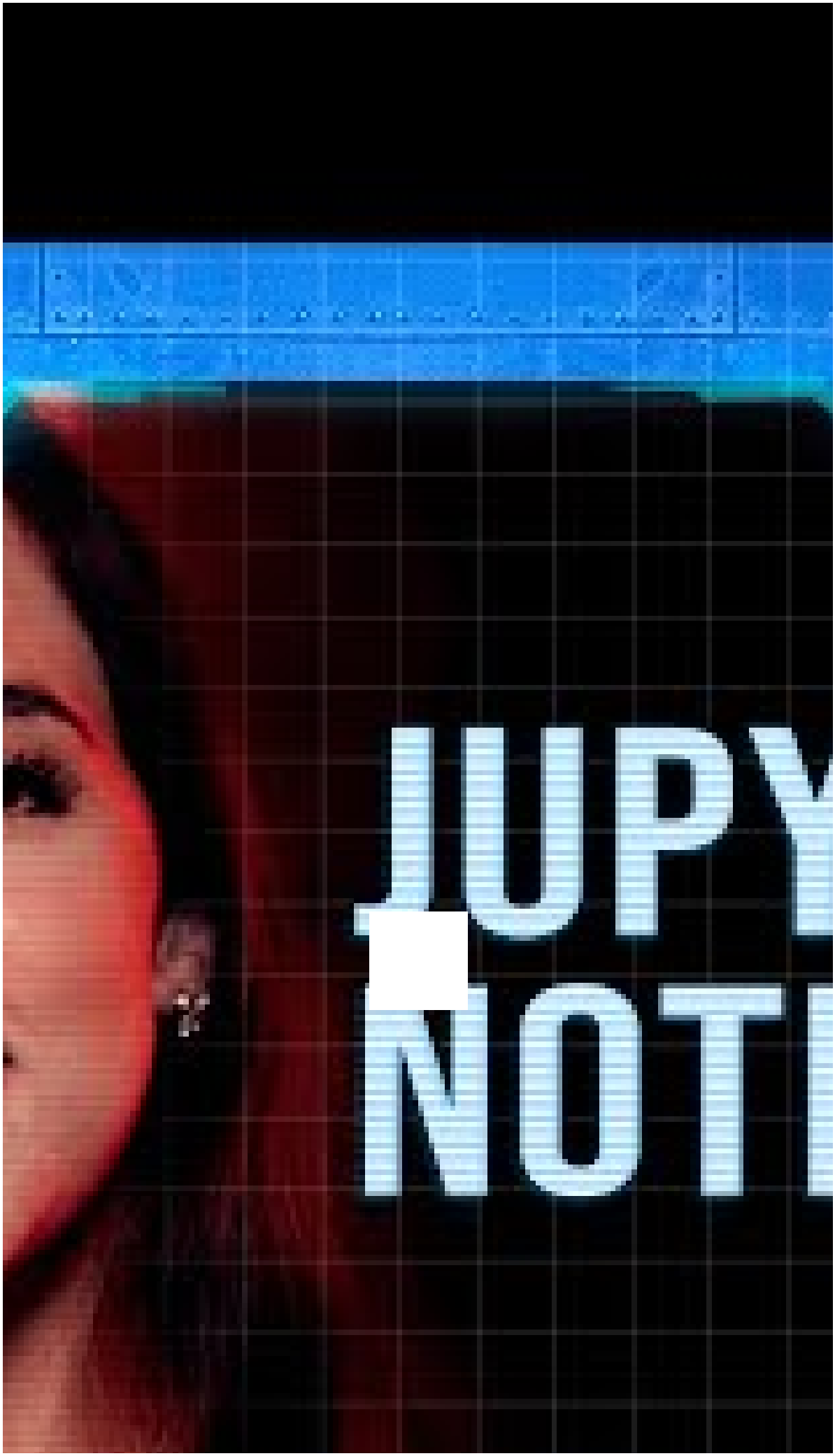
explorando a praticidade do modo interativo, enquanto se escreve código e já observa prontamente a sua saída, conforme a seguinte imagem:

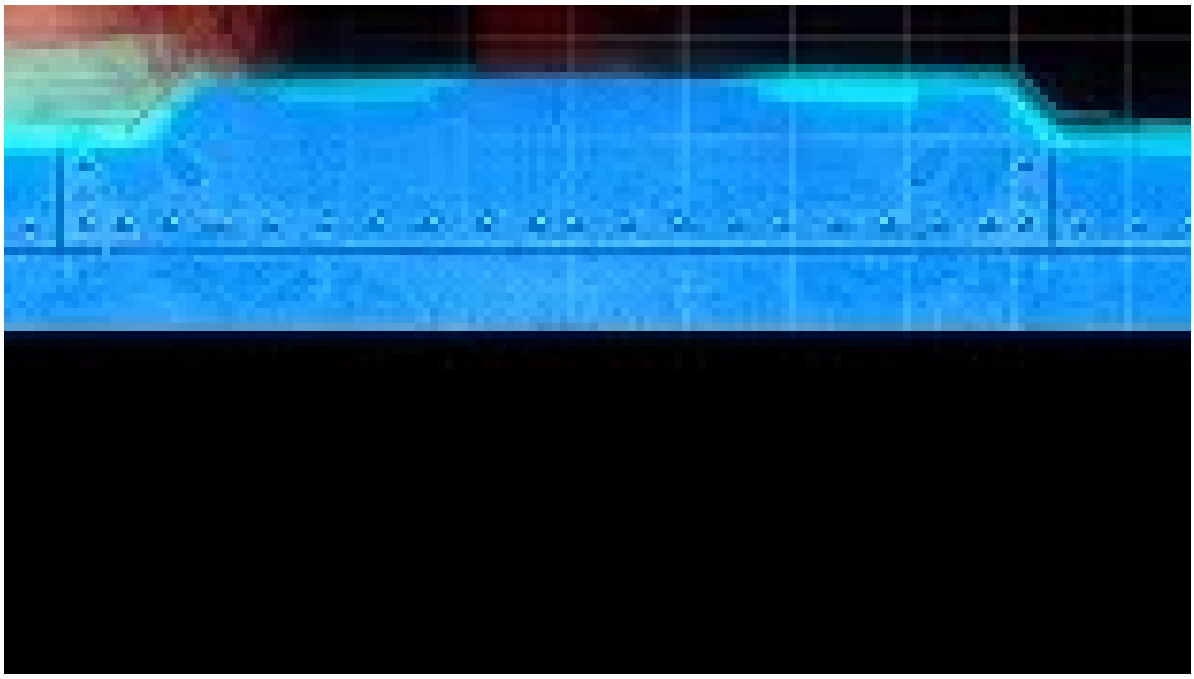


Além dos Jupyter Notebooks, também é possível trabalhar com scripts Python comuns (arquivos .py). A diferença é que a saída de todos os fragmentos de código é colocada no terminal sem distinção, uma após a outra, e em formato *raw* (cru). O exemplo abaixo mostra como seria a mesma saída, em um script equivalente, no terminal:



Nesse episódio do Hipsters Ponto Tube, a cientista de dados Mikaeri Ohana conversa com o Paulo Silveira, CEO da Alura, sobre como uma pessoa Cientista de Dados utiliza a ferramenta Jupyter Notebook no dia a dia.


JUPY
NOTE



Instalação do Pandas

A maneira mais fácil e simples de instalar, segundo a própria documentação do Pandas, é instalando a **distribuição do Anaconda**.

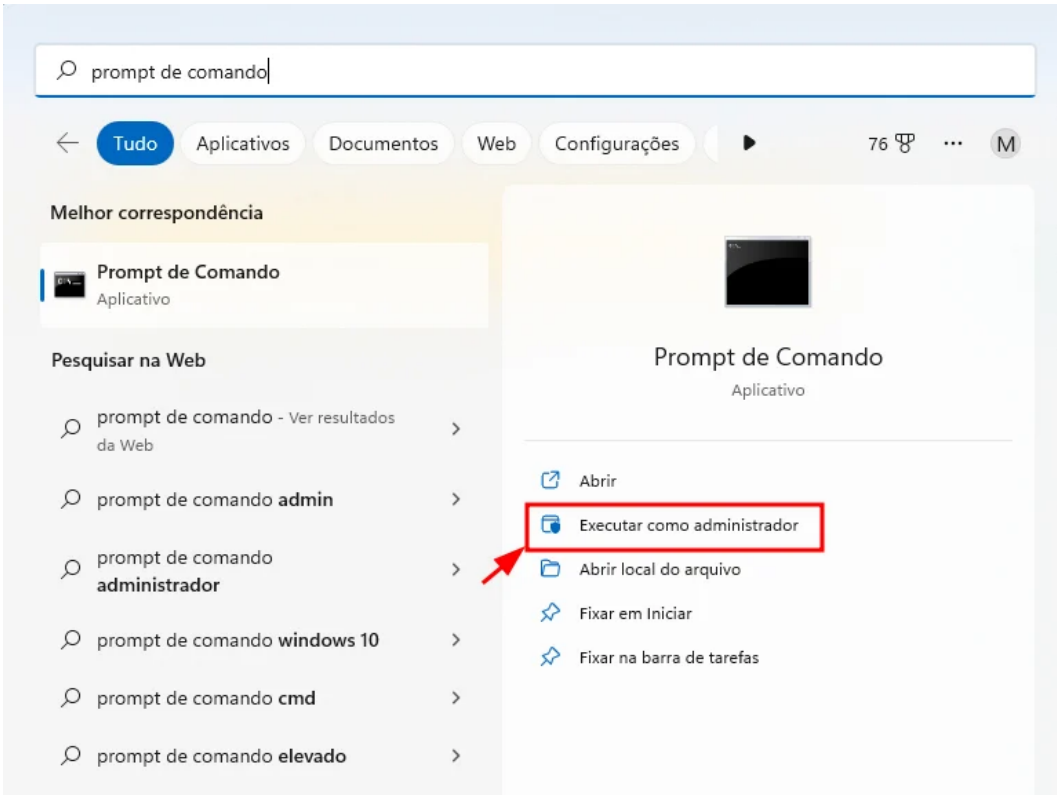
O Anaconda é um ambiente de desenvolvimento voltado para Ciência de Dados com Python e R, que trás instaladas várias bibliotecas e softwares de uso popular no ramo. Dentre as bibliotecas instaladas, temos também o Pandas. Você pode aprender como instalar o Anaconda no Windows através da documentação oficial do Anaconda.

Uma outra maneira comum de instalar o Pandas é utilizando o **PIP**, o sistema de gerenciamento de pacotes do Python.

Desde que você tenha feito o download do Python a partir do site oficial, podemos utilizar o seguinte procedimento:

> ***Atenção**: Caso você tenha mais de um disco rígido na sua máquina, é preciso garantir que a instalação está sendo feita no mesmo disco onde o Python foi instalado.*

1) Para começar, devemos abrir o Prompt de Comando do seu sistema operacional. No Windows, pressione as teclas de atalho Windows + R, digite "Prompt de Comando", e clique na opção "Executar como administrador":



2) O Prompt de Comando será aberto e surgirá a tela preta do terminal. Nesse momento, podemos verificar a versão do Python instalada na máquina com o comando `python --version` e garantir que podemos continuar:

```
python --version
```

```
Python 3.9.7
```

3) Caso você ainda não tenha o PIP instalado na máquina, pode instalá-lo utilizando um módulo nativo do Python para isso, com o comando:

```
python -m ensurepip --upgrade
```

4) E, agora que já temos o PIP instalado na máquina, podemos utilizá-lo para instalar o Pandas, com o comando:

```
pip install pandas
```

5) Pronto, agora nós já temos o Pandas instalado na máquina.

Conclusão

Se você deseja mergulhar ainda mais nos conteúdos de Pandas e Ciência de Dados, aqui na Alura nós temos a Formação Python para Data Science. A formação aborda as principais ferramentas utilizadas em Ciência de Dados com Python, tais como Pandas, Numpy, Matplotlib, Seaborn, e muito mais. Nela, construímos vários projetos práticos para compor o seu portfólio como profissional de dados.

E se você já deu seus primeiros passos nessa ferramenta, te convidamos a participar dos Challenges de Data Science. Neles, você pode trabalhar na construção de um portfólio de projetos, desenvolvendo habilidades em limpeza, tratamento e visualização de dados, e também competências em Machine Learning.

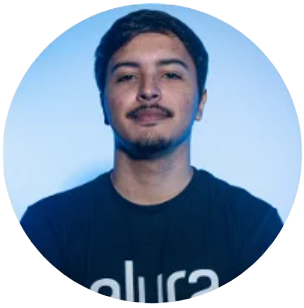
Créditos:

Conteúdo: Marcus Almeida

Produção técnica: Rodrigo Dias

Produção didática: Thaís de Faria

Designer gráfico: Alysson Manso



Marcus Almeida

Bacharelando em Engenharia Elétrica pelo Instituto Federal do Maranhão. Atuou como parte do Scuba Team da Escola de Dados na Alura, trabalhando com conteúdos voltados a Data Science, Machine Learning, Python e SQL. Adora conversar tecnologia, universo geek, games e também aprender coisas novas.

Artigo Anterior

Engenharia de Dados: o que é, o que faz e um Guia completo

Próximo Artigo

Kubernetes: conhecendo a orquestração de containers

Leia também:

- Engenharia de Dados: o que é, o que faz e um Guia completo
- Power BI: o que é, para que serve, quem utiliza e um Guia para iniciar
- Saiba tudo sobre SQL - A linguagem padrão para trabalhar com banco de dados relacionais!
- Por onde começar os estudos na área de dados?
- O que é Ciências de Dados?
- ML Engineer, o Dev em <T>

Veja outros artigos sobre
Data Science

Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

ME INSCREVA

Nossas redes e apps

Institucional

Sobre nós

Trabalhe conosco

Para Empresas

Para Escolas

Política de Privacidade

A Alura

Formações

Como Funciona

Todos os cursos

Depoimentos

Instrutores(as)

Cursos de DevOps
AWS | Azure | Docker | Segurança | IaC | Linux

Cursos de UX & Design
Usabilidade e UX | Vídeo e Motion | 3D

Cursos de Mobile
React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão
Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas