

# Biblioteca

# NumPy / Pandas

# **Integrantes**

**Gabriel Leonardo Vicente Cancian N° 02**

**Gabriel Portocarrero de Oliveira N° 03**

**Guilherme Ballestrim Sobreira N°06**

**João Vitor Alves da Costa n°11**

**Matheus Lima Catarucci N° 20**

**Moisés Gabriel Tafarello N° 22**

# Introdução

**Neste trabalho, vamos conhecer duas bibliotecas muito usadas no Python: o Pandas e o NumPy. Vamos entender para que elas servem, como funcionam e ver exemplos práticos de como podem ajudar no dia a dia, especialmente quando trabalhamos com análise e manipulação de dados.**

# Conceito de biblioteca

## As bibliotecas no python

são módulos que agrupam funções e algoritmos prontos, facilitando o desenvolvimento do código.

## Exemplo de uso

```
import math
```

```
print(math.sqrt(25)) # Saída: 5.0
```

- Biblioteca que já vem no python
- Função sqrt()

# Bibliotecas locais x Externas

## As bibliotecas padrão

são as bibliotecas que já vem integradas por padrão no python.

### Exemplos:

- math
- random
- datetime

## As bibliotecas não padrão

são as bibliotecas que não vem integradas por padrão no python.

### Exemplos:

- pandas
- numpy
- flask



# Biblioteca

# NumPy

# NumPy – Biblioteca para Cálculos com Arrays

**NumPy significa Numerical Python**

**NumPy é uma biblioteca Python usada para trabalhar com matrizes.**

**Ela também possui funções para trabalhar no domínio da álgebra linear, transformada de Fourier e matrizes.**

**NumPy foi criado em 2005 por Travis Oliphant.**

# NumPy – Biblioteca para Cálculos

O que é o NumPy?

Trabalha com arrays N-dimensionais

Muito mais rápido e eficiente que listas Python

Usado por bibliotecas como Pandas, Scikit-Learn, etc.

Está se tornando uma biblioteca “padrão”

“ndarray”



# NumPy – Biblioteca para Cálculos com Arrays

O que é um Array?

Um array Python nada mais é do que uma coleção ou compilação de uma série de elementos de um determinado tipo de dado que são armazenados juntos.

É a famosa “lista” que já aprendemos.

# NumPy – Biblioteca para Cálculos com Arrays

**Arrays do NumPy são N-dimensionais**

**Arrays podem ter 1, 2, 3... até N dimensões**

**Estrutura mais poderosa que listas do Python**

**Permitem cálculos matemáticos rápidos e vetorizados**

**São usados em dados complexos (ex: imagens, vídeos)**

# NumPy – Biblioteca para Cálculos com Arrays

**Arrays do NumPy são N-dimensionais**

Os arrays do NumPy são armazenados em um único lugar contínuo na memória, diferentemente das listas, então os processos podem acessá-los e manipulá-los de forma muito eficiente.

Processo chamado de: “Localidade de Referência”

# Vantagens do NumPy

## Bem mais rápido que listas

Diferente das listas comuns do Python, os arrays do NumPy são mais rápidos e consomem menos memória, tornando-o uma escolha ideal para manipulação de grandes volumes de dados numéricos.

## Operações vetoriais simples

O NumPy permite a criação de vetores n-dimensionais, também conhecidos como tensores. Essas estruturas podem representar dados em diversas dimensões, sendo ideais para manipulação de imagens, dados volumosos em machine learning e simulações físicas complexas.

## Compatibilidade

O NumPy é amplamente compatível e se integra de forma eficiente com bibliotecas populares do ecossistema Python, como Pandas (análise de dados), SciPy (computação científica) e TensorFlow (aprendizado de máquina), servindo como base para operações numéricas de alto desempenho.



# Desvantagens do NumPy

## Tipos de dados homogêneos

matrizes NumPy são homogêneas, o que significa que só podem conter elementos do mesmo tipo de dados. Isso pode ser problemático ao trabalhar com conjuntos de dados que possuem vários tipos de dados.

## Capacidades limitadas de manipulação de strings

O NumPy não possui recursos robustos de manipulação de strings. Ele só consegue lidar com strings de comprimento fixo, o que pode ser limitante ao trabalhar com dados de texto.

## Curva de aprendizado íngreme

O NumPy pode ser desafiador para iniciantes. Ele tem uma API grande e complexa, e sua sintaxe pode ser confusa para usuários não familiarizados com conceitos de computação científica.



# Biblioteca

# Pandas

# Biblioteca Pandas

## O que é o pandas?

O pandas é uma biblioteca para trabalhar com ciência de dados, open source. O pandas pode ser usado para limpeza e tratamento de dados, análise exploratória de dados, etc.



# Pontos positivos

## Sintaxe de fácil entendimento

```
import pandas as pd

# Criando o DataFrame
dados = {
    'Nome': ['Ana', 'Bruno', 'Carlos'],
    'Idade': [22, 35, 28]
}
df = pd.DataFrame(dados)

print(df)
```

**Ampla documentação e tutoriais na internet**  
repositório no github

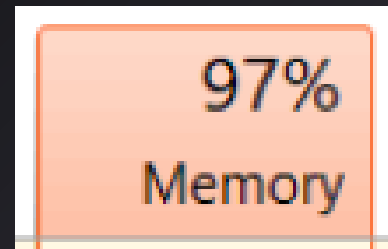
**Fácil integração com outras bibliotecas**

NumPy é um ótimo exemplo



# Pontos negativos

**Alto uso de memória RAM com grandes volumes de dados**



→ Travamentos

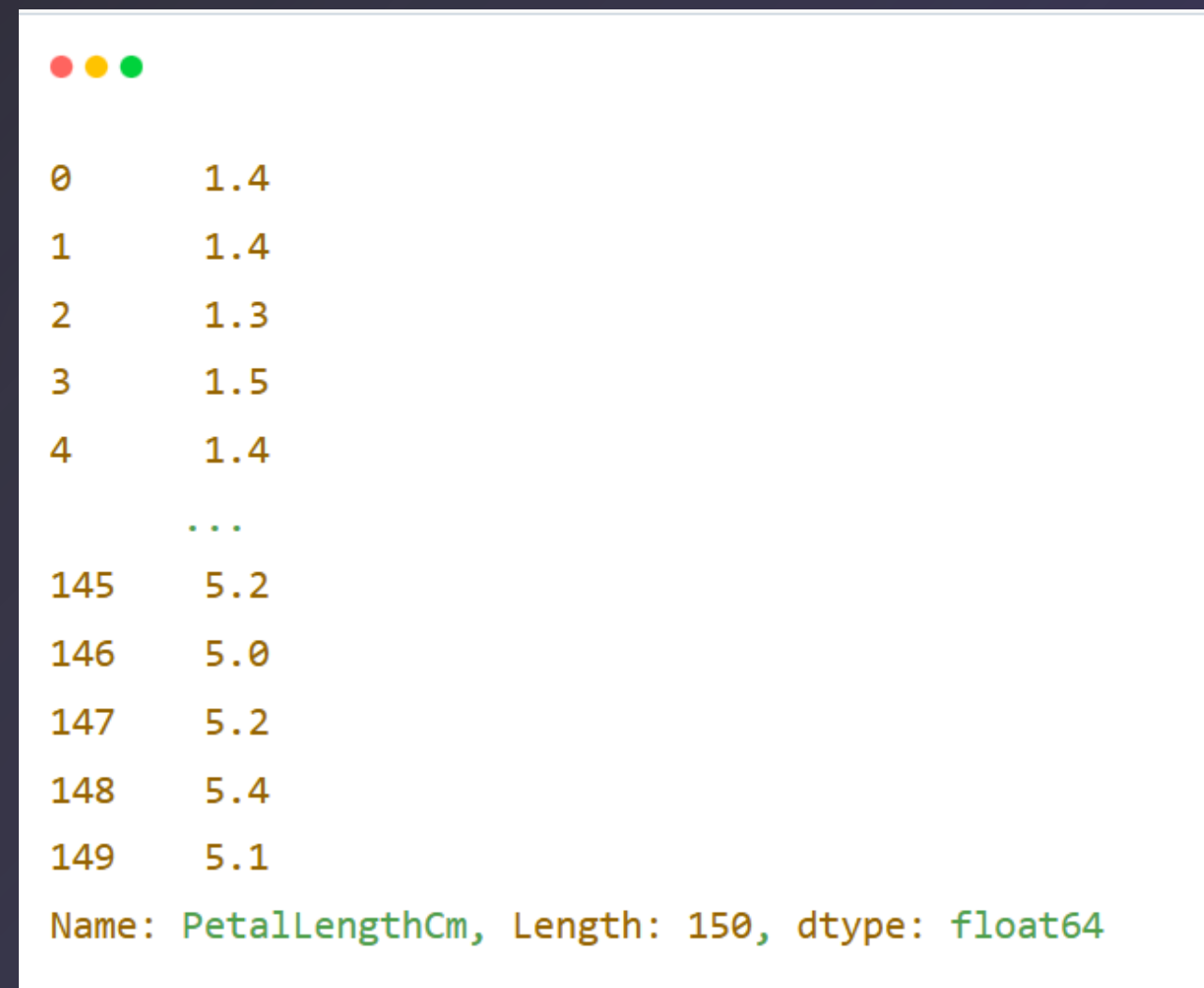
**Não ideal para bigdata**

Para grandes volumes de dados são recomendados outras bibliotecas, como spark e dask

# Estruturas do panda

## Series (Coluna)

As Series são objetos de tipo array unidimensional, com um eixo de rótulos, também chamado de index, que é responsável por identificar cada registro.



```
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 terminal window with a white background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays a pandas Series of floating-point values. The first five rows show indices 0 to 4 with values 1.4, 1.4, 1.3, 1.5, and 1.4. An ellipsis indicates intermediate rows. The last five rows show indices 145 to 149 with values 5.2, 5.0, 5.2, 5.4, and 5.1. At the bottom, the series metadata is shown: 'Name: PetalLengthCm, Length: 150, dtype: float64'.



# Series

```
import pandas as pd

idades = pd.Series([22, 35], index=['Ana', 'Bruno'])
print(idades)
input()
```

Saída -->

```
Ana      22
Bruno    35
dtype: int64
```

# Estruturas do panda

## DataFrame (Tabela completa)

Os DataFrames são objetos bidimensionais, de tamanho variável. O seu formato é de uma tabela, onde os dados são organizados em linhas e colunas.

|     | 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 |

# DataFrame

```
import pandas as pd

dados = {'Nome': ['Ana', 'Bruno'], 'Idade': [22, 35]}
df = pd.DataFrame(dados)
print(df)
input()
```

Saída -->

|   | Nome  | Idade |
|---|-------|-------|
| 0 | Ana   | 22    |
| 1 | Bruno | 35    |

# Conclusão

As bibliotecas pandas e numpy são realmente incríveis e facilitam muito a vida de quem trabalha com dados. O NumPy é ótimo para lidar com números e cálculos rápidos, enquanto o

Pandas ajuda a organizar e analisar dados de um jeito bem prático, como se fossem tabelas. O melhor de tudo é que eles funcionam super bem juntos, tornando o processo de analisar e manipular informações muito mais fácil e rápido. Por isso, aprender a usar essas duas ferramentas faz toda a diferença para quem quer trabalhar com dados de verdade.



# Referências

## ALURA PANDAS

alura.com.br/artigos/numpy-computacao-cientifica-com-python?  
srsltid=AfmBOoqxRsKkGJE8DzuVJR1mhvO1FgK3YMqDKz6mHG7BB\_OKEhL4pUd

## ALURA NUMPY

alura.com.br/artigos/pandas-o-que-e-para-que-serve-como-instalar?srsltid=AfmBOoownLyV-  
RHpZQj8c7kxE4JvvEpRbihJHhOraComsJkQ3IdOITnz

## DOCUMENTAÇÃO OFICIAL DO NUMPY

<https://numpy.org/doc/stable/>



Obrigado pela  
atenção!