

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**

# **Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn**

## **Pandas – Ler e escrever**

Aula 3

Código da aula: [DADOS]ANO1C2B3S24A3

**Bibliotecas: Pandas,  
NumPy, SciPy,  
Matplotlib e Seaborn**

## Mapa da Unidade 5 Componente 3

NumPy – Manipulação  
de array

semana

22

Pandas – Acesso e  
seleção

semana

23

semana

20

NumPy: acesso,  
inspeção e índice

semana

21

NumPy – Operação  
de arrays

semana

24

**Você está aqui!**  
Pandas – Ler e  
escrever

**Bibliotecas: Pandas,  
NumPy, SciPy,  
Matplotlib e Seaborn**

## **Mapa da Unidade 5 Componente 3**

# **Você está aqui!**

**Pandas – Ler e escrever**

**24**

### **Aula 3**

Código da aula:  
[DADOS]ANO1C2B3S24A3



## Objetivos da aula

- Entender o conceito de ler ou carregar um arquivo na biblioteca Pandas no Python.



## Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Acesso ao laboratório de informática e/ou à internet.
- Software Anaconda/Jupyter Notebook instalado ou similar.



## Duração da aula

50 minutos



## Competências técnicas

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados.
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências.



## Competências socioemocionais

- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados.
- Trabalhar em equipes multifuncionais, colaborando com colegas, gestores e clientes.



Ponto de partida

# Ler um arquivo no Pandas

Imagine que temos estes arquivos. De que forma podemos acessá-los, carregá-los e lê-los como dataframe no Pandas?

📁	dados	
📄	[DADOS]C3U5S24A1A2A3A4.ipynb	13.8 KB
📄	cad_alunos.csv	118 B
📄	comidas.csv	437 B
📄	cores.csv	96 B
📄	futebol.csv	178 B
📄	linguagens.csv	137 B
📄	meu_dataframe_sem_cabecalho.csv	80 B
📄	meu_dataframe_sem_indice.csv	91 B
📄	meu_dataframe.csv	100 B
📄	meu_dataframe.xlsx	4.9 KB
📄	paises.csv	166 B
📄	praias.csv	294 B

Elaborado especialmente para o curso.

# Construindo o conceito

## Pandas

Em resumo, podemos criar um DataFrame a partir de um dicionário. Existem outras formas de criar um DataFrame, e você pode explorar na documentação oficial do Pandas: <https://pandas.pydata.org/>.

```
1 import pandas as pd
2
3 cores = {
4     "Nome": ["Vermelho", "Azul", "Verde", "Amarelo", "Roxo"],
5     "Código Hex": ["#FF0000", "#0000FF", "#00FF00", "#FFFF00", "#800080"]
6 }
7 df_cores = pd.DataFrame(cores)
8 df_cores.to_csv("cores.csv", index=False)
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Para salvar, usamos o método `to_csv` ou `to_excel`. Mas como ler esses arquivos?

## Leitura de diversos tipos de arquivos



### Ler um arquivo CSV

O método `pd.read_csv()` é usado para ler dados de um arquivo CSV em um DataFrame do Pandas.



### Ler um arquivo JSON

Use o método `pd.read_json()` para ler dados de um arquivo JSON em um DataFrame do Pandas.



### Ler um arquivo Excel

O método `pd.read_excel()` é usado para ler dados de um arquivo Excel (formato XLSX) em um DataFrame.



### Ler um arquivo HTML

Use o método `pd.read_html()` para ler tabelas contidas em uma página HTML. Ele retorna uma lista de DataFrames (uma para cada tabela encontrada na página).



### Ler um arquivo de texto

Para ler dados de um arquivo de texto (como um arquivo TXT), você pode usar o método `pd.read_csv()` com o argumento `sep`, especificando o separador correto.



Construindo  
o **conceito**

# Leitura de diversos tipos de arquivos

Exemplos:

```
1 # Lê dados de um arquivo CSV em um DataFrame
2 df = pd.read_csv('meu_arquivo.csv')
3
4 # Lê dados de um arquivo Excel em um DataFrame
5 df = pd.read_excel('meu_arquivo.xlsx')
6
7 # Lê dados de um arquivo de texto em um DataFrame
8 df = pd.read_csv('meu_arquivo.txt', sep='\t') # Supondo que o separador seja o TAB
9
10 # Lê dados de um arquivo JSON em um DataFrame
11 df_json = pd.read_json('meu_arquivo.json')
12
13 # Lê tabelas de uma página HTML em uma lista de DataFrames
14 lista_df_html = pd.read_html('minha_pagina.html')
15
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

## Construindo o conceito

# Leitura de diversos tipos de arquivos

Usando os arquivos salvos da aula passada:

- países.csv
- países.xlsx
- países.json

```
1 # Lê dados de um arquivo CSV em um DataFrame
2 df_países = pd.read_csv('países.csv')
3 df_países
```

	Nome	Continente	População (milhões)
0	Brasil	América do Sul	212
1	Estados Unidos	América do Norte	331
2	França	Europa	67
3	Japão	Ásia	126
4	Austrália	Oceania	25

```
1 # Lê dados de um arquivo Excel em um DataFrame
2 df_países_excel = pd.read_excel('países.xlsx')
3 df_países_excel
```

	Nome	Continente	População (milhões)
0	Brasil	América do Sul	212
1	Estados Unidos	América do Norte	331
2	França	Europa	67
3	Japão	Ásia	126
4	Austrália	Oceania	25

```
1 # Lê dados de um arquivo JSON em um DataFrame
2 df_países_json = pd.read_json('países.json')
3 df_países_json
```

	Nome	Continente	População (milhões)
0	Brasil	América do Sul	212
1	Estados Unidos	América do Norte	331
2	França	Europa	67
3	Japão	Ásia	126
4	Austrália	Oceania	25

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

## Construindo o conceito

# Leitura de diversos tipos de arquivos

Usando os arquivos salvos da aula passada:

- países.html
- países.pickle

```
1 # Lê tabelas de uma página HTML em uma lista de DataFrames
2 df_html = pd.read_html('países.html')
3 df_html[0]
```

	Nome	Continente	População (milhões)
0	Brasil	América do Sul	212
1	Estados Unidos	América do Norte	331
2	França	Europa	67
3	Japão	Ásia	126
4	Austrália	Oceania	25

```
1 # Lê arquivo binário no formato pickle
2 df_pickle = pd.read_pickle('países.pkl')
3 df_pickle
```

	Nome	Continente	População (milhões)
0	Brasil	América do Sul	212
1	Estados Unidos	América do Norte	331
2	França	Europa	67
3	Japão	Ásia	126
4	Austrália	Oceania	25

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

## Construindo o conceito

# Pandas – read\_csv()

O método **read\_csv()** é usado para ler dados de arquivos CSV (Comma-Separated Values) em um DataFrame do Pandas.

Aqui estão algumas informações sobre como usá-lo:

```
1 import pandas as pd
2
3 # Lê dados de um arquivo CSV em um DataFrame
4 df = pd.read_csv('meu_arquivo.csv')
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

### Argumentos:

**filepath\_or\_buffer:** Especifica o caminho do arquivo CSV (pode ser um arquivo local ou uma URL).

**sep:** Define o caractere usado como separador entre os valores no arquivo CSV (padrão é a vírgula ,).

**header:** Controla se a primeira linha contém os nomes das colunas (padrão é inferir os nomes das colunas).

## Construindo o conceito

# Pandas – read\_csv()

Usando alguns argumentos para abrir os arquivos salvos na aula anterior:

```
1 pd.read_csv('alunos_com_separador.csv')
```

Nome Idade Curso		
0	Ana 20 Engenharia	
1	Carlos 22 Administração	
2	Maria 21 Medicina	
3	Pedro 19 Ciência da Computação	

```
1 pd.read_csv('alunos_com_separador.csv', sep='|')
```

	Nome	Idade	Curso
0	Ana	20	Engenharia
1	Carlos	22	Administração
2	Maria	21	Medicina
3	Pedro	19	Ciência da Computação

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



## Construindo o conceito

# Pandas – read\_csv()

Podemos ler também arquivos csv diretamente de uma URL (endereço da web).

```
1 pd.read_csv("https://raw.githubusercontent.com/remarchese/seduc/main/arquivos/paises.csv")
```

	Nome	Continente	População (milhões)
0	Brasil	América do Sul	212
1	Estados Unidos	América do Norte	331
2	França	Europa	67
3	Japão	Ásia	126
4	Austrália	Oceania	25

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



© Getty Images

O que nós  
**aprendemos  
hoje?**

## Então ficamos assim...

- 1** Conhecemos vários métodos da biblioteca Pandas no Python que permite ler arquivos;
- 2** Conhecemos `read_csv`, `read_excel`, `read_json`, `read_html` que são alguns exemplos de métodos de leitura no Pandas;
- 3** Vimos também que é possível ler um arquivo diretamente da URL.

# Saiba mais

Sabia que o Pandas se integra com o SQL, abrindo um leque de possibilidades para sua análise de dados? Descubra como usar o poder do SQL dentro do Pandas e explore um universo de insights!

ALURA. *Python*: análise de dados com SQL. 08 Importando a SQLAlchemy. Disponível em: <https://cursos.alura.com.br/course/python-analise-dados-sql/task/134360>.

Acesso em: 3 jul. 2024.



# Referências da aula

MCKINNEY, W. *Python para análise de dados: tratamento de dados com Pandas, NumPy e Jupyter*. São Paulo: Novatec, 2023.

PANDAS. *Página inicial*, [s.d.]. Disponível em: <https://pandas.pydata.org/>. Acesso em: 3 jul. 2024.

Identidade visual: Imagens © Getty Images.

**Educação  
Profissional  
Paulista**

Técnico em  
**Ciência de  
Dados**