

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**

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

## **Pandas – Ler e escrever**

Aula 2

Código da aula: [DADOS]ANO1C2B3S24A2

**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 2**

Código da aula: [DADOS]ANO1C2B3S24A2



## Objetivos da aula

- Praticar o conceito de salvar arquivos de diferentes tipos na biblioteca Pandas do 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.



# Construindo o conceito

## Pandas – Salvar arquivos

```
1 import pandas as pd
2
3 # Dicionário com informações sobre a população de alguns países
4 populacao = {
5     "País": ["China", "Índia", "Estados Unidos", "Indonésia", "Paquistão", "Brasil", "Nigéria", "Bangladesh"],
6     "População (milhões)": [1441, 1393, 332, 276, 225, 213, 211, 166]
7 }
8
9 # Criando o DataFrame a partir do dicionário
10 df_populacao = pd.DataFrame(populacao)
11
12 df_populacao
```

	País	População (milhões)
0	China	1441
1	Índia	1393
2	Estados Unidos	332
3	Indonésia	276
4	Paquistão	225
5	Brasil	213
6	Nigéria	211
7	Bangladesh	166

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Vimos que um dataframe do Pandas pode ser salvo em csv e xlsx.

Porém, há outros formatos que o Pandas pode salvar.

Usando o dataframe da imagem, vamos salvar em outras extensões.

## Construindo o conceito

# Pandas – Salvar arquivos

```
1 import pandas as pd
2
3 # Dicionário com informações sobre a população de alguns países
4 populacao = {
5     "País": ["China", "Índia", "Estados Unidos", "Indonésia", "Paquistão", "Brasil", "Nigéria", "Bangladesh"],
6     "População (milhões)": [1441, 1393, 332, 276, 225, 213, 211, 166]
7 }
8
9 # Criando o DataFrame a partir do dicionário
10 df_populacao = pd.DataFrame(populacao)
11
12 # Salvando em diferentes formatos
13 df_populacao.to_csv('populacao.csv', index=False)
14 df_populacao.to_excel('populacao.xlsx', index=False)
15 df_populacao.to_json('populacao.json', orient='records')
16 df_populacao.to_html('populacao.html', index=False)
17
18 conexao = 'sqlite:///meu_banco_de_dados.db'
19 df_populacao.to_sql('tabela_populacao', con=conexao, if_exists='replace', index=False)
20
21 df_populacao.to_pickle('populacao.pkl')
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

# Construindo o **conceito**

Métodos para salvar arquivos	
<b>to_csv()</b>	Este método salva o DataFrame em um arquivo CSV (Comma-Separated Values).
<b>to_excel()</b>	Este método salva o DataFrame em um arquivo Excel (formato XLSX).
<b>to_json()</b>	Este método salva o DataFrame em um arquivo JSON.
<b>to_html()</b>	Este método salva o DataFrame em um arquivo HTML.
<b>to_sql()</b>	Este método permite salvar um DataFrame em uma tabela SQL (banco de dados relacional).
<b>to_pickle()</b>	Este método salva o DataFrame em um arquivo binário no formato pickle. O formato pickle é específico do Python e pode ser usado para armazenar objetos Python.

Elaborado especialmente para o curso.



# Construindo o conceito

## Pandas – Salvar arquivos

Compare os diferentes arquivos:

-  populacao.pkl
-  meu\_banco\_de\_dados.db
-  populacao.html
-  populacao.json
-  populacao.xlsx
-  populacao.csv

	Pais	Populacao (milhoes)
1	China	1441
2	India	1393
3	Estados Unidos	332
4	Indonesia	276
5	Paquistao	225
6	Brasil	213
7	Nigeria	211
8	Bangladesh	166

populacao.csv

Pais	Populacao (milhoes)
China	1441
India	1393
Estados Unidos	332
Indonesia	276
Paquistao	225
Brasil	213
Nigeria	211
Bangladesh	166

populacao.html

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

# Construindo o conceito

## Pandas – Salvar arquivos

Compare os diferentes arquivos:

	A	B
1	<b>Pais</b>	<b>Populacao (milhoes)</b>
2	China	1441
3	India	1393
4	Estados Unidos	332
5	Indonesia	276
6	Paquistao	225
7	Brasil	213
8	Nigeria	211
9	Bangladesh	166
10		

populacao.xlsx

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

```
1  [  
2      {  
3          "Pais": "China",  
4          "Populacao (milhoes)": 1441  
5      },  
6      {  
7          "Pais": "India",  
8          "Populacao (milhoes)": 1393  
9      },  
10     {  
11         "Pais": "Estados Unidos",  
12         "Populacao (milhoes)": 332  
13     },  
14     {  
15         "Pais": "Indonesia",  
16         "Populacao (milhoes)": 276  
17     },  
18     {  
19         "Pais": "Paquistao",  
20         "Populacao (milhoes)": 225  
21     },  
22     {  
23         "Pais": "Brasil",  
24         "Populacao (milhoes)": 213  
25     },  
26     {  
27         "Pais": "Nigeria",  
28         "Populacao (milhoes)": 211  
29     },  
30     {  
31         "Pais": "Bangladesh",  
32         "Populacao (milhoes)": 166  
33     }  
34 ]
```

populacao.json

## Colocando em **prática**

# Exercícios



**Durante a aula**



**Em grupo**

**1**

Crie dataframes a partir de dicionário e salve-os (sem índice) em arquivos com extensões: csv, xlsx, html, json e pickle

**2**

Salve com o nome cores.

```
cores = { "Nome": ["Vermelho", "Azul", "Verde", "Amarelo", "Roxo"], "Código Hex": ["#FF0000", "#0000FF", "#00FF00", "#FFFF00", "#800080"] }
```

**3**

Salve com o nome alunos

```
alunos = { "Nome": ["Ana", "Carlos", "Maria", "Pedro"], "Idade": [20, 22, 21, 19], "Curso": ["Engenharia", "Administração", "Medicina", "Ciência da Computação"] }
```

**4**

Salve com o nome paises

```
paises = { "Nome": ["Brasil", "Estados Unidos", "França", "Japão", "Austrália"], "Continente": ["América do Sul", "América do Norte", "Europa", "Ásia", "Oceania"], "População (milhões)": [212, 331, 67, 126, 25] }
```

Colocando  
em **prática**

## Exercícios



**Durante a aula**



**Em grupo**

**5**

Crie dataframes a partir de dicionário e salve-os (sem índice) em arquivos com extensões: csv, xlsx, html, json e pickle

**6**

Salve com o nome linguagens.

```
linguagens = { "Nome": ["Python", "Java", "JavaScript", "C++",  
"Ruby"], "Ano de Criação": [1991, 1995, 1995, 1983, 1995],  
"Popularidade": ["Muito alta", "Alta", "Muito alta", "Alta", "Média"] }
```

**7**

Salve com o nome alunos

```
comidas = { "Nome": ["Pizza", "Sushi", "Lasanha", "Churrasco",  
"Chocolate"], "Descrição": [ "Uma deliciosa massa coberta com  
molho de tomate, queijo e diversos ingredientes.", "Prato  
japonês feito com arroz temperado, peixe cru e outros  
acompanhamentos.", "Camadas de massa intercaladas com  
molho de carne, queijo e molho branco.", "Carne assada na  
brasa, típica de churrascarias brasileiras.", "Doce feito a partir  
do cacau, disponível em várias formas e sabores." ] }
```

# Construindo o **conceito**

Argumentos do método <code>to_csv()</code>		Exemplos
<b>path_or_buf</b>	Especifica o caminho do arquivo em que o DataFrame será salvo. Pode ser uma string com o nome do arquivo ou um objeto de arquivo (como um objeto <code>io.StringIO</code> ).	<code>df.to_csv('meu_dataframe.csv')</code>
<b>index</b>	Controla se o índice do DataFrame será incluído no arquivo CSV. O valor padrão é <code>True</code> . Para excluir o índice, use <code>index=False</code> .	<code>df.to_csv('meu_dataframe.csv', index=False)</code>
<b>sep</b>	Define o caractere usado como separador entre os valores no arquivo CSV. O valor padrão é a vírgula ( <code>,</code> ).	<code>df.to_csv('meu_dataframe.csv', sep=';')</code>
<b>header</b>	Controla se o cabeçalho (nomes das colunas) será incluído no arquivo CSV. O valor padrão é <code>True</code> . Para excluir o cabeçalho, use <code>header=False</code> .	<code>df.to_csv('meu_dataframe.csv', header=False)</code>
<b>encoding</b>	Define a codificação do arquivo CSV (por exemplo, <code>'utf-8'</code> , <code>'latin-1'</code> etc.). O valor padrão é <code>'utf-8'</code> .	<code>df.to_csv('meu_dataframe.csv', encoding='latin-1')</code>

Elaborado especialmente para o curso.



Construindo  
o **conceito**

## Argumentos do método `to_csv()`

Exemplo:

```
1 import pandas as pd
2
3 # Criando um DataFrame de exemplo
4 dados = {
5     "Nome": ["Ana", "Carlos", "Maria", "Pedro"],
6     "Idade": [20, 22, 21, 19],
7     "Curso": ["Engenharia", "Administração", "Medicina", "Ciência da Computação"]
8 }
9 df_alunos = pd.DataFrame(dados)
10
11 # Salvando em diferentes formatos
12 df_alunos.to_csv('alunos_sem_indice.csv', index=False) # Sem índice
13 df_alunos.to_csv('alunos_com_separador.csv', sep='|', index=False) # Separado por "|"
14 df_alunos.to_csv('alunos_com_encoding.csv', encoding='latin1', index=False) # Com encoding "latin1"
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



**Dica:**

Abra os arquivos e explore o resultado.



© Getty Images

O que nós  
**aprendemos**  
**hoje?**

## Então ficamos assim...

- 1** Aprendemos que a biblioteca Pandas pode salvar um dataframe em diferentes tipos de arquivos;
- 2** Vimos que é possível salvar arquivos csv, xlsx, json, html, entre outros;
- 3** Compreendemos que os argumentos dos métodos para salvar um arquivo permitem manipular melhor a forma do arquivo na hora de salvar.

# Saiba mais

Está com a memória um pouco enferrujada sobre as estruturas de dados do Pandas? Recupere seus conhecimentos e domine essa biblioteca essencial para análise de dados!

ALURA. *Pandas: conhecendo a biblioteca*. 09 Estruturas de dados do Pandas. Disponível em: <https://cursos.alura.com.br/course/pandas-conhecendo-biblioteca/task/126147>.

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**