

# Lição 1: Introdução ao Pandas e Configuração do Ambiente – Analisando Dados de Times de Futebol

Nesta primeira lição, você será introduzido ao **Pandas**, uma das bibliotecas mais poderosas para análise de dados em Python. Para tornar o aprendizado mais interessante, usaremos **estatísticas de times de futebol**, como **gols marcados, vitórias, empates, derrotas e saldo de gols**.

Ao longo da lição, aprenderemos a **instalar o Pandas, entender suas estruturas principais (Series e DataFrames), carregar dados de arquivos CSV e explorar um ambiente interativo como Jupyter Notebook ou Google Colab**.

## 1.1 O que é o Pandas e por que utilizá-lo?

O **Pandas** é uma biblioteca essencial para quem trabalha com dados estruturados, oferecendo ferramentas para **filtragem, manipulação, análise e visualização de informações**. Ele é amplamente utilizado em diversas áreas, incluindo **ciência de dados, estatística, economia e esportes**.

A principal característica do Pandas é a sua capacidade de lidar com dados **tabulares** (como planilhas do Excel, tabelas SQL ou arquivos CSV), permitindo manipulação eficiente por meio de suas estruturas de dados **Series** e **DataFrame**.

O nome **Pandas** vem de "**Panel Data**", um termo estatístico que se refere a conjuntos de dados multidimensionais usados para análise econômica.

### Por que usar Pandas para análise de dados?

- ✓ **Facilidade para manipular tabelas e estatísticas** (pontos, gols, vitórias)
- ✓ **Leitura e gravação de arquivos CSV, Excel, JSON**
- ✓ **Funções matemáticas e estatísticas embutidas**
- ✓ **Integração com Matplotlib e Seaborn para criar gráficos**
- ✓ **Rápida e eficiente manipulação de grandes conjuntos de dados**

Ao longo deste material, você verá como o Pandas pode **ajudar a analisar e comparar times, prever tendências e gerar estatísticas detalhadas**.

## 1.2 Instalando e Configurando o Pandas

Antes de começarmos a trabalhar com dados de futebol, precisamos instalar o Pandas no nosso ambiente Python.

### Instalação com pip

Abra o terminal ou prompt de comando e digite:

```
pip install pandas
```

Para verificar a instalação, execute:

```
import pandas as pd  
print(pd.__version__)
```

Se a versão do Pandas for exibida, significa que tudo está funcionando corretamente!

### Usando Jupyter Notebook ou Google Colab

O **Jupyter Notebook** é um ambiente interativo que permite executar código Python de forma mais organizada. Para instalá-lo, use:

```
pip install notebook
```

Para rodar o Jupyter:

```
jupyter notebook
```

Caso prefira não instalar nada, o **Google Colab** é uma excelente alternativa baseada na nuvem. Basta acessar: [Google Colab](#)

### **DICA RÁPIDA** 🏆:

*Prefira o Google Colab se quiser um ambiente pronto para uso sem precisar configurar nada no seu computador.*

## 1.3 Estruturas do Pandas: Series e DataFrame

O Pandas oferece duas estruturas principais de dados:

### 1.3.1 Series – Dados Unidimensionais

Uma **Series** pode ser vista como uma **coluna única de dados** com um índice associado. No contexto do futebol, podemos representar a quantidade de **gols marcados por um time nas últimas rodadas**.

```
gols = pd.Series([2, 1, 3, 0, 4], index=['Rodada 1', 'Rodada 2', 'Rodada 3', 'Rodada 4', 'Rodada 5'])
print(gols)
print(gols)
```

**Saída:**

```
Rodada 1    2
Rodada 2    1
Rodada 3    3
Rodada 4    0
Rodada 5    4
```

### Exemplo 2: Gols Sofridos

Imagine que você queira representar a quantidade de gols sofridos por um time nas últimas rodadas.

```
import pandas as pd
```

```
gols_sofridos = pd.Series([1, 0, 2, 1, 3],
                          index=['R1', 'R2', 'R3', 'R4', 'R5'])
print(gols_sofridos)
```

**Saída:**

```
Rodada 1    1
Rodada 2    0
Rodada 3    2
Rodada 4    1
Rodada 5    3
```

### Exemplo 3: Assistências

Agora, veja um exemplo em que a Series armazena a quantidade de assistências realizadas pelo time em cada rodada.

```
import pandas as pd
```

```
assistencias = pd.Series([0, 2, 1, 3, 2],
                          index=['Rodada 1', 'Rodada 2', 'Rodada 3', 'Rodada 4',
                                'Rodada 5'])
print(assistencias)
```

**Saída:**

```
Rodada 1    0
Rodada 2    2
Rodada 3    1
Rodada 4    3
Rodada 5    2
```

### 1.3.2 DataFrame – Estrutura Tabular

Um **DataFrame** é uma **tabela bidimensional** composta por **várias Series**. Podemos representar um campeonato fictício com dados de vitórias, empates e derrotas:

```
dados = {
    'Time': ['Flamengo', 'Palmeiras', 'São Paulo', 'Corinthians'],
    'Vitórias': [18, 16, 14, 12],
    'Empates': [5, 7, 9, 10],
    'Derrotas': [5, 6, 7, 8]
}
```

```
df = pd.DataFrame(dados)
print(df)
```

**Saída:**

	Time	Vitórias	Empates	Derrotas
0	Flamengo	18	5	5
1	Palmeiras	16	7	6
2	São Paulo	14	9	7
3	Corinthians	12	10	8

## Exemplo 2: Tabela de Artilheiros

Neste exemplo, criamos um DataFrame com informações dos principais artilheiros do campeonato, contendo o nome do jogador, o time, a quantidade de gols e assistências:

```
import pandas as pd
```

```
dados_artilheiros = {  
    'Jogador': ['Gabriel Barbosa', 'Dudu', 'Marinho', 'Gustavo Scarpa'],  
    'Time': ['Flamengo', 'Palmeiras', 'São Paulo', 'Corinthians'],  
    'Gols': [22, 18, 15, 12],  
    'Assistências': [5, 7, 6, 4]  
}
```

```
df_artilheiros = pd.DataFrame(dados_artilheiros)  
print(df_artilheiros)
```

### Saída:

	Jogador	Time	Gols	Assistências
0	Gabriel Barbosa	Flamengo	22	5
1	Dudu	Palmeiras	18	7
2	Marinho	São Paulo	15	6
3	Gustavo Scarpa	Corinthians	12	4

## Exemplo 3: Estatísticas de Partidas

Neste exemplo, criamos um DataFrame com dados de desempenho dos times em cada rodada, contendo a rodada, o time, a posse de bola (em %), número de finalizações e chutes a gol:

```
import pandas as pd
```

```
dados_partidas = {
```

```

'Rodada': ['Rodada 1', 'Rodada 2', 'Rodada 3', 'Rodada 4'],
'Time': ['Flamengo', 'Palmeiras', 'São Paulo', 'Corinthians'],
'Posse de Bola (%)': [58, 55, 52, 60],
'Finalizações': [15, 13, 11, 14],
'Chutes a Gol': [7, 5, 4, 6]
}

```

```

df_partidas = pd.DataFrame(dados_partidas)
print(df_partidas)

```

**Saída:**

	Rodada	Time	Posse de Bola (%)	Finalizações	Chutes a Gol
0	Rodada 1	Flamengo	58	15	7
1	Rodada 2	Palmeiras	55	13	5
2	Rodada 3	São Paulo	52	11	4
3	Rodada 4	Corinthians	60	14	6

**DICA RÁPIDA** 🏆:

Um **DataFrame** pode ser comparado a uma **tabela do Excel** ou a um **banco de dados SQL**. Ele facilita a análise e manipulação de dados esportivos!

## 1.4 Lendo e Gravando Arquivos de Dados

### Lendo um arquivo CSV

Podemos carregar arquivos CSV com estatísticas de futebol:

```

df = pd.read_csv('times_brasileirao.csv')
print(df.head()) # Exibe as primeiras 5 linhas

```

### Gravando um arquivo CSV

Após manipularmos os dados, podemos salvá-los:

```

df.to_csv('estatisticas_times.csv', index=False)

```

## Resumo da Lição 1

- ✓ O Pandas é uma biblioteca para manipulação de dados estruturados
- ✓ Possui duas estruturas principais: **Series** (coluna única) e **DataFrame** (tabela)
- ✓ Podemos criar DataFrames manualmente ou carregar dados de arquivos
- ✓ O Pandas é amplamente utilizado em análises esportivas, incluindo estatísticas de times de futebol

## Exercícios Práticos

- 1 - **Crie uma Series** representando a quantidade de gols marcados pelo seu time favorito nas últimas 5 partidas.
- 2 - **Crie um DataFrame** com 4 times do Brasileirão e suas estatísticas de vitórias, empates e derrotas.
- 3 - **Salve o DataFrame em um arquivo CSV** e carregue-o novamente.

Se precisar de ajuda, volte nesta lição e revise os exemplos!