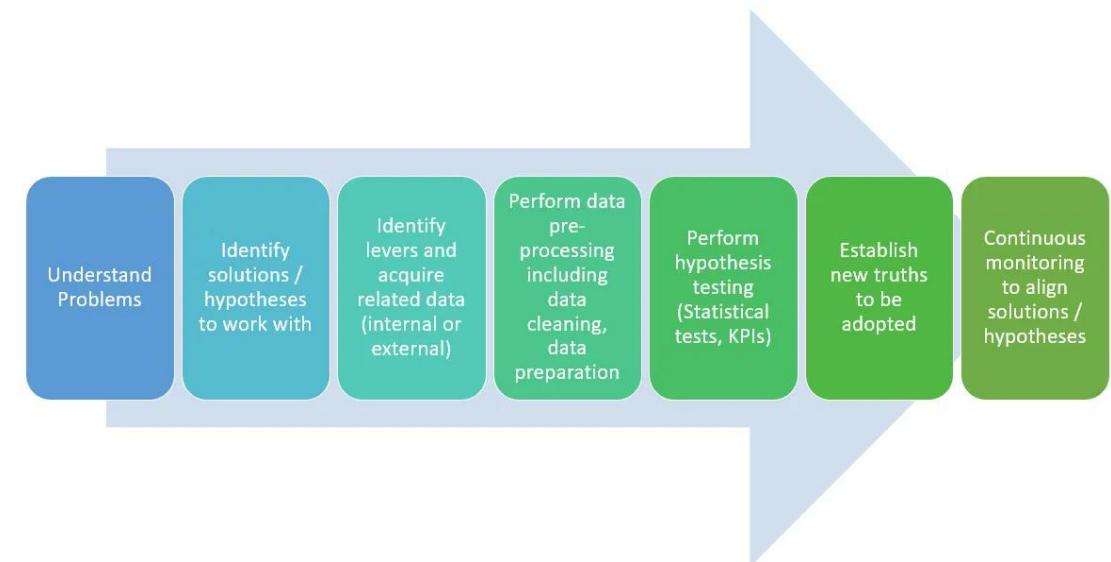


Ciência de Dados no Nosso Dia a Dia

A **Ciência de Dados** está presente em praticamente todos os aspectos da nossa vida digital. É a área que combina estatística, programação e conhecimento de domínio para extrair insights valiosos de grandes volumes de dados.

- **Netflix** usa algoritmos para analisar seu histórico de visualização e recomendar filmes e séries que você provavelmente vai gostar.
- **Anúncios online** são personalizados com base no seu comportamento de navegação, histórico de compras e interesses detectados.
- **Spotify** analisa seus hábitos de escuta para criar playlists personalizadas e recomendar novas músicas.
- **Python** é a linguagem de programação mais utilizada em Ciência de Dados devido à sua simplicidade e poderosas bibliotecas.



Recursos Essenciais de Python: Listas e Dicionários

O Python possui estruturas de dados poderosas que são fundamentais para a Ciência de Dados:

```
In [147]: df = data1.explode('event_params').reset_index(drop = True)
df['key'] = None

for i in range(len(df)):
    df.loc[i, 'key'] = df.loc[i, 'event_params']['key']

df['value'] = None

for i in range(len(df)):
    for k in df.loc[i, 'event_params']['value']:
        if df.loc[i, 'event_params']['value'][k] != None:
            df.loc[i, 'value'] = df.loc[i, 'event_params']['value'][k]

df.drop(columns = 'event_params', inplace = True)

-----
TypeError                                 Traceback (most recent call last)
<ipython-input-147-69344af5a8d4> in <module>
      4
      5 for i in range(len(df)):
----> 6     df.loc[i, 'key'] = df.loc[i, 'event_params']['key']
      7
      8 df['value'] = None

TypeError: string indices must be integers
```

• Listas

São como [caixas](#) onde podemos guardar vários itens em ordem. Cada item tem uma posição (índice) que começa em 0.

```
# Criando uma lista
frutas = ["maçã", "banana", "laranja"]
```

```
# Acessando elementos pelo índice
print(frutas[0]) # Primeiro elemento
```

maçã

• Dicionários

São como [agendas de contatos](#), onde cada nome (chave) tem um número (valor) associado.

```
# Criando um dicionário
estoque = {"maçã": 15, "banana": 8}
```

```
# Acessando valores pela chave
print(f'Estoque de maçãs: {estoque["maçã"]}')
```

Estoque de maçãs: 15

Recurso Essencial de Python: Pandas DataFrame

O [Pandas](#) é a biblioteca mais importante para análise de dados em Python. Sua principal estrutura de dados é o [DataFrame](#):

• DataFrame

É como uma tabela de planilha (Excel) com linhas e colunas, perfeita para organizar e analisar dados de forma estruturada.

- Manipulação eficiente de grandes conjuntos de dados
- Funções para limpeza, transformação e análise de dados
- Integração com outras bibliotecas de visualização

```
import pandas as pd

# Criando um DataFrame
dados_alunos = {
    "Nome": ["Ana", "Pedro", "Maria"],
    "Idade": [25, 30, 22],
    "Cidade": ["São Paulo", "Rio de Janeiro", "Belo Horizonte"]
}

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

	Nome	Idade	Cidade
0	Ana	25	São Paulo
1	Pedro	30	Rio de Janeiro
2	Maria	22	Belo Horizonte

```
In [147]: df = data1.explode('event_params').reset_index(drop = True)

df['key'] = None

for i in range(len(df)):
    df.loc[i, 'key'] = df.loc[i, 'event_params']['key']

df['value'] = None

for i in range(len(df)):
    for k in df.loc[i, 'event_params']['value']:
        if df.loc[i, 'event_params']['value'][k] != None:
            df.loc[i, 'value'] = df.loc[i, 'event_params']['value'][k]

df.drop(columns = 'event_params', inplace = True)

-----
TypeError                                         Traceback (most recent call last)
<ipython-input-147-69344af5a8d4> in <module>
      4
      5 for i in range(len(df)):
----> 6     df.loc[i, 'key'] = df.loc[i, 'event_params']['key']
      7
      8 df['value'] = None

TypeError: string indices must be integers
```

Google Colab: Seu Laboratório Online de Python

O [Google Colab](#) (Colaboratory) é um ambiente de notebook baseado na nuvem que permite escrever e executar código Python diretamente no navegador, sem necessidade de configuração.



Baseado na Nuvem

Não precisa instalar nada no seu computador. Basta ter uma conta Google e um navegador para começar a programar.



Bibliotecas Pré-instaladas

Vem com as principais bibliotecas de Ciência de Dados já instaladas: Pandas, NumPy, Matplotlib, Scikit-learn e muitas outras.



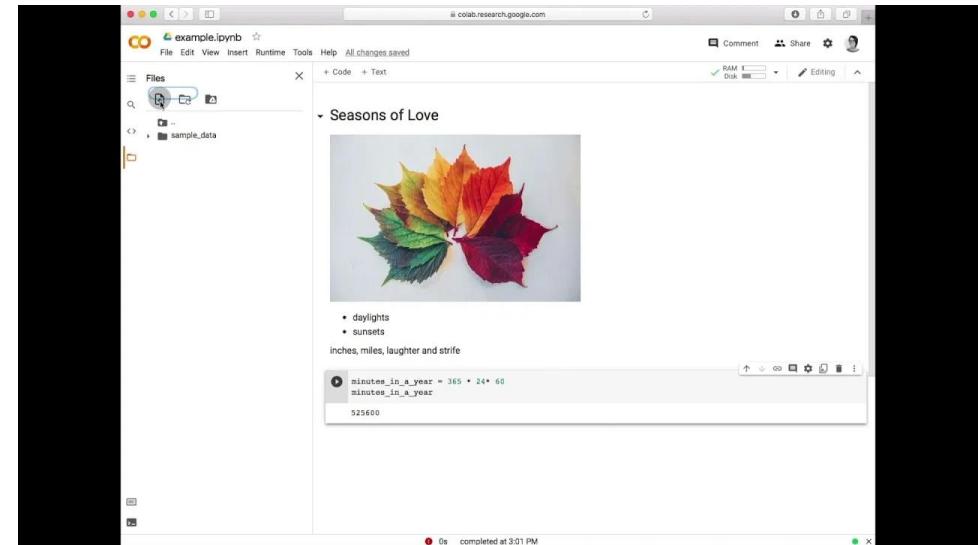
Recursos Computacionais Gratuitos

Oferece acesso gratuito a GPUs e TPUs, permitindo executar algoritmos complexos sem precisar de hardware potente.



Fácil Compartilhamento

Permite compartilhar seus notebooks com outras pessoas, facilitando a colaboração em projetos e o ensino.



Atividade A: Conectando e Explorando o Colab

Vamos começar nossa jornada prática com o [Google Colab](#). Siga os passos abaixo para configurar seu ambiente de trabalho:

1 Acessando o Colab

Abra seu navegador e acesse o endereço colab.research.google.com. Você precisará fazer login com sua conta Google.

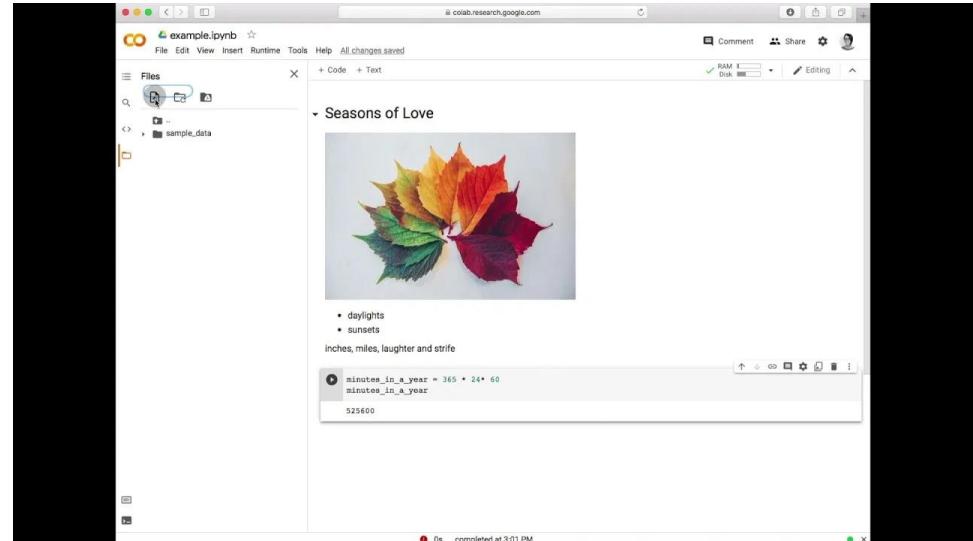
2 Criando o Notebook

Clique em **File (Arquivo) > New Notebook (Novo Notebook)** para criar um novo documento de trabalho.

3 Explorando a Interface

Familiarize-se com os elementos principais:

- Adicione uma célula de código clicando em **+ Code**
- Adicione uma célula de texto clicando em **+ Text**
- Execute o código clicando no ícone de "Play" ou usando **Shift + Enter**



4 Importando as Bibliotecas

Na primeira célula de código, digite e execute:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Estas são as bibliotecas essenciais para Ciência de Dados:

- **pd** - Pandas para manipulação de dados
- **np** - NumPy para cálculos numéricos
- **plt** - Matplotlib para visualizações

Atividade B: Exercício Individual – Criando Estruturas de Dados

Neste exercício individual, você vai praticar a criação de diferentes estruturas de dados no Google Colab. Continue no notebook que você criou anteriormente:

1 Crie uma Lista

Em uma célula de código, digite e execute:

```
frutas = ["maçã", "banana", "laranja", "morango"]
print(frutas)
```

2 Crie um Dicionário

Em outra célula, digite e execute:

```
estoque = {"maçã": 15, "banana": 8, "laranja": 12}
print(f'O estoque de maçãs é de: {estoque["maçã"]}')
```

3 Crie um DataFrame do Pandas

Em uma nova célula, digite e execute:

```
dados_alunos = {
    "Nome": ["Ana", "Pedro", "Maria"],
    "Idade": [25, 30, 22],
    "Cidade": ["São Paulo", "Rio de Janeiro", "Belo
Horizonte"]
}

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

```
In [147]: df = data1.explode('event_params').reset_index(drop = True)

df['key'] = None

for i in range(len(df)):
    df.loc[i, 'key'] = df.loc[i, 'event_params']['key']

df['value'] = None

for i in range(len(df)):
    for k in df.loc[i, 'event_params']['value']:
        if df.loc[i, 'event_params']['value'][k] != None:
            df.loc[i, 'value'] = df.loc[i, 'event_params']['value'][k]

df.drop(columns = 'event_params', inplace = True)

-----
TypeError                                         Traceback (most recent call last)
<ipython-input-147-69344af5a8d4> in <module>
      4
      5 for i in range(len(df)):
----> 6     df.loc[i, 'key'] = df.loc[i, 'event_params']['key']
      7
      8 df['value'] = None

TypeError: string indices must be integers
```

Atividade C: Atividade em Dupla – Criando Gráficos

Nesta atividade, vocês trabalharão em [duplas](#) para criar gráficos usando NumPy e Matplotlib.

1 Criando Dados com NumPy

Em uma nova célula de código:

```
# Criando um array NumPy com notas
notas = np.array([85, 90, 78, 92, 88])
```

2 Criando um Gráfico de Linha

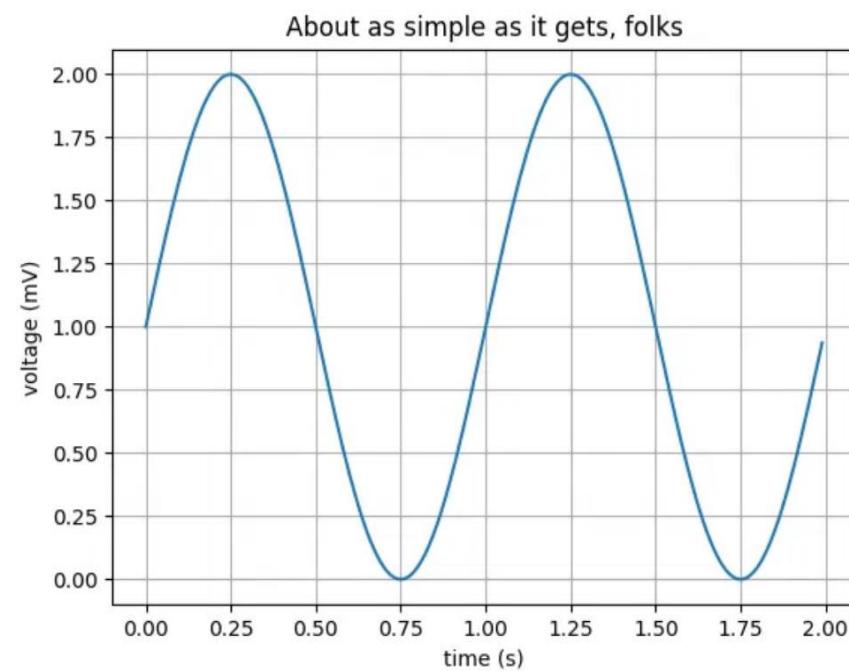
Em outra célula:

```
plt.figure(figsize=(8, 5))
plt.plot(notas)
plt.title("Notas de Prova")
plt.xlabel("Alunos")
plt.ylabel("Notas")
plt.show()
```

3 Criando um Gráfico de Barras

Vamos criar um gráfico de barras:

```
plt.figure(figsize=(8, 5))
alunos = ["Ana", "Pedro", "Maria", "João", "Lucas"]
plt.bar(alunos, notas)
plt.title("Notas por Aluno")
plt.show()
```



Atividade D: Explorando o Editor do Colab

Vamos explorar algumas funcionalidades do editor do Google Colab:

1 Comentários no Código

Adicione anotações no seu código usando o símbolo #:

```
# Este é um comentário explicando o código  
notas = [85, 90, 78] # Lista com as notas
```

2 Células de Texto (Markdown)

Adicione uma célula de texto clicando em **+ Text**:

```
# Título Principal  
## Subtítulo  
  
**Texto em negrito** e *texto em itálico*
```

3 Renomear o Notebook

Dê um nome significativo ao seu notebook:

- Clique no nome atual no topo da página
- Digite um novo nome como "Analise_Inicial.ipynb"

