

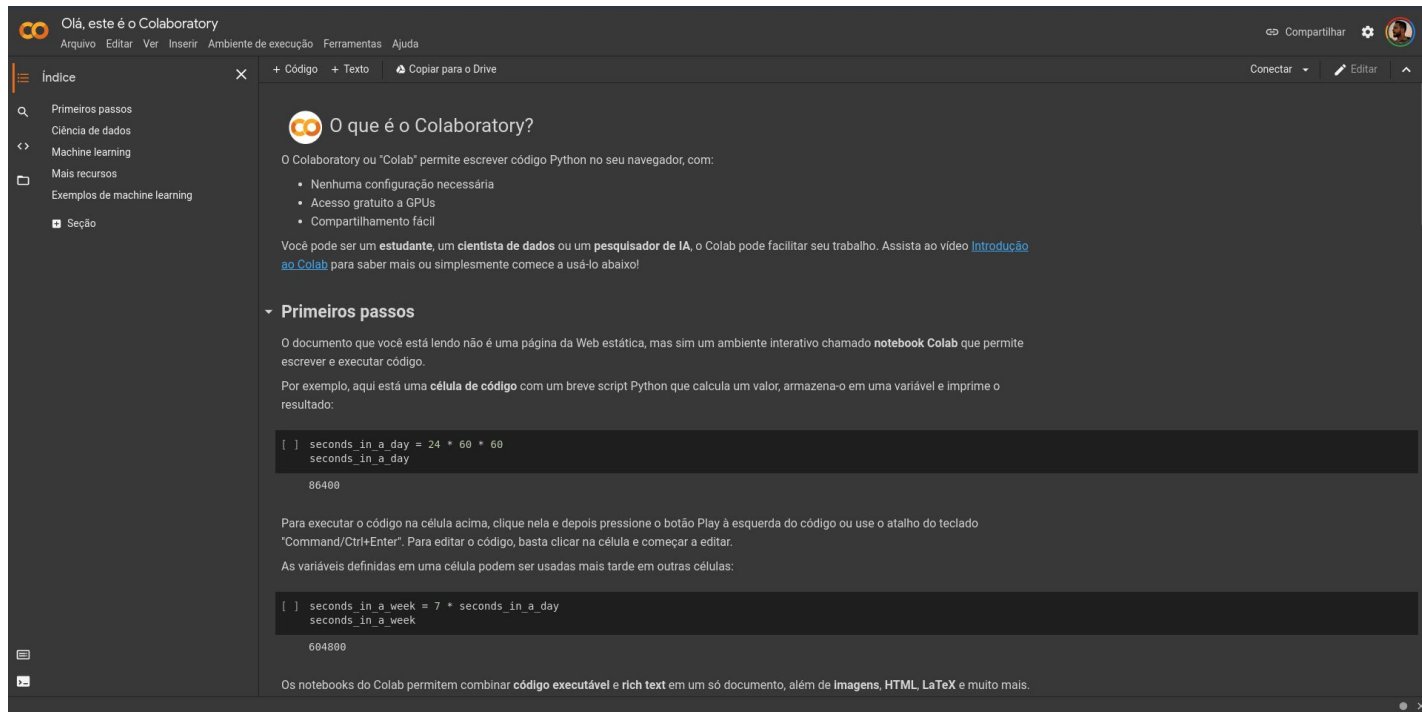
A large red square with a white border, centered on a white background. Inside the square, the text "Tutorial" and "Google Colab" are written in white.

# Tutorial

## Google Colab

# Acessando a plataforma

## Welcome To Colaboratory - Colaboratory



The screenshot displays the Google Colaboratory web interface. At the top, a dark header bar contains the Colab logo, the text 'Olá, este é o Colaboratory', and a menu with options: 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. On the right side of the header are links for 'Compartilhar', a settings gear icon, and a user profile icon. Below the header, a left sidebar shows a table of contents under the heading 'Índice', including 'Primeiros passos', 'Ciência de dados', 'Machine learning', 'Mais recursos', 'Exemplos de machine learning', and 'Seção'. The main content area has a dark background and features the Colab logo and the title 'O que é o Colaboratory?'. It explains that Colab allows writing Python code in a browser and lists three benefits: no configuration needed, free GPU access, and easy sharing. It also provides a link to an 'Introdução ao Colab' video. A section titled 'Primeiros passos' describes the interactive 'Colab notebook' environment. It includes a code cell with a Python script to calculate seconds in a day, which has been executed, showing the output '86400'. Below this, it explains how to run code by clicking the cell or pressing 'Command/Ctrl+Enter', and how to edit code by clicking the cell. It also mentions that variables defined in one cell can be used in others. Another code cell is shown with a script to calculate seconds in a week, resulting in the output '604800'. The bottom of the page states that Colab notebooks can combine executable code and rich text in a single document, supporting images, HTML, and LaTeX.

Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Compartilhar ⚙️ 👤

Índice

- Primeiros passos
- Ciência de dados
- Machine learning
- Mais recursos
- Exemplos de machine learning
- Seção

+ Código + Texto Copiar para o Drive

Conectar Editar

### O que é o Colaboratory?

O Colaboratory ou "Colab" permite escrever código Python no seu navegador, com:

- Nenhuma configuração necessária
- Acesso gratuito a GPUs
- Compartilhamento fácil

Você pode ser um **estudante**, um **cientista de dados** ou um **pesquisador de IA**, o Colab pode facilitar seu trabalho. Assista ao vídeo [Introdução ao Colab](#) para saber mais ou simplesmente comece a usá-lo abaixo!

#### Primeiros passos

O documento que você está lendo não é uma página da Web estática, mas sim um ambiente interativo chamado **notebook Colab** que permite escrever e executar código.

Por exemplo, aqui está uma **célula de código** com um breve script Python que calcula um valor, armazena-o em uma variável e imprime o resultado:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day

86400
```

Para executar o código na célula acima, clique nela e depois pressione o botão Play à esquerda do código ou use o atalho do teclado "Command/Ctrl+Enter". Para editar o código, basta clicar na célula e começar a editar.

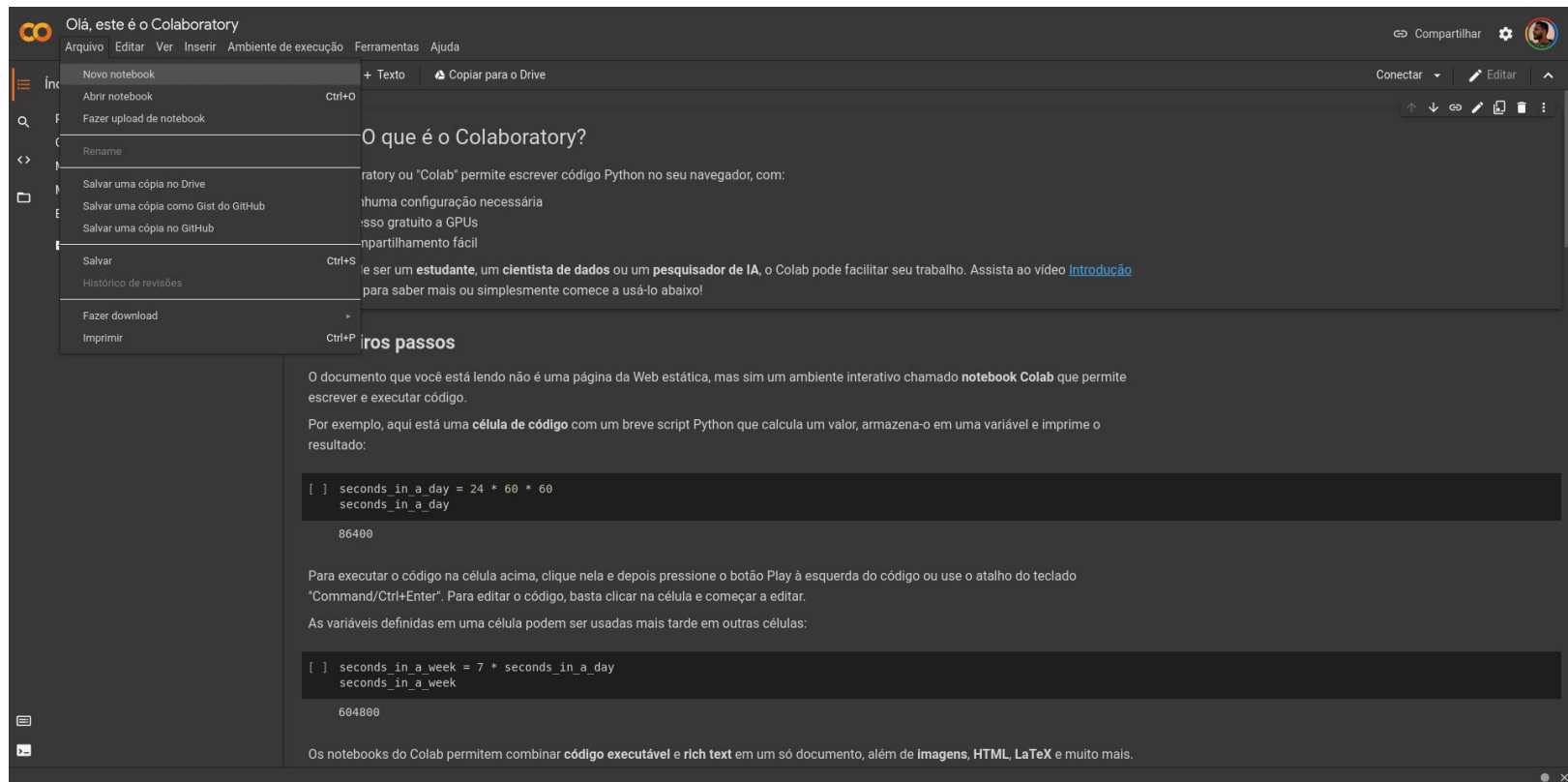
As variáveis definidas em uma célula podem ser usadas mais tarde em outras células:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
    seconds_in_a_week

604800
```

Os notebooks do Colab permitem combinar **código executável** e **rich text** em um só documento, além de **imagens**, **HTML**, **LaTeX** e muito mais.

# Criando um novo notebook



Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Novo notebook + Texto Copiar para o Drive

Abzir notebook Ctrl+O

Fazer upload de notebook

Rename

Salvar uma cópia no Drive

Salvar uma cópia como Gist do GitHub

Salvar uma cópia no GitHub

Salvar Ctrl+S

Histórico de revisões

Fazer download

Imprimir Ctrl+P

O que é o Colaboratory?

Colaboratory ou "Colab" permite escrever código Python no seu navegador, com:  
- nenhuma configuração necessária  
- acesso gratuito a GPUs  
- compartilhamento fácil

Se você é um **estudante**, um **cientista de dados** ou um **pesquisador de IA**, o Colab pode facilitar seu trabalho. Assista ao vídeo [Introdução](#) para saber mais ou simplesmente comece a usá-lo abaixo!

## Os passos

O documento que você está lendo não é uma página da Web estática, mas sim um ambiente interativo chamado **notebook Colab** que permite escrever e executar código.

Por exemplo, aqui está uma **célula de código** com um breve script Python que calcula um valor, armazena-o em uma variável e imprime o resultado:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

86400

Para executar o código na célula acima, clique nela e depois pressione o botão Play à esquerda do código ou use o atalho do teclado "Command/Ctrl+Enter". Para editar o código, basta clicar na célula e começar a editar.

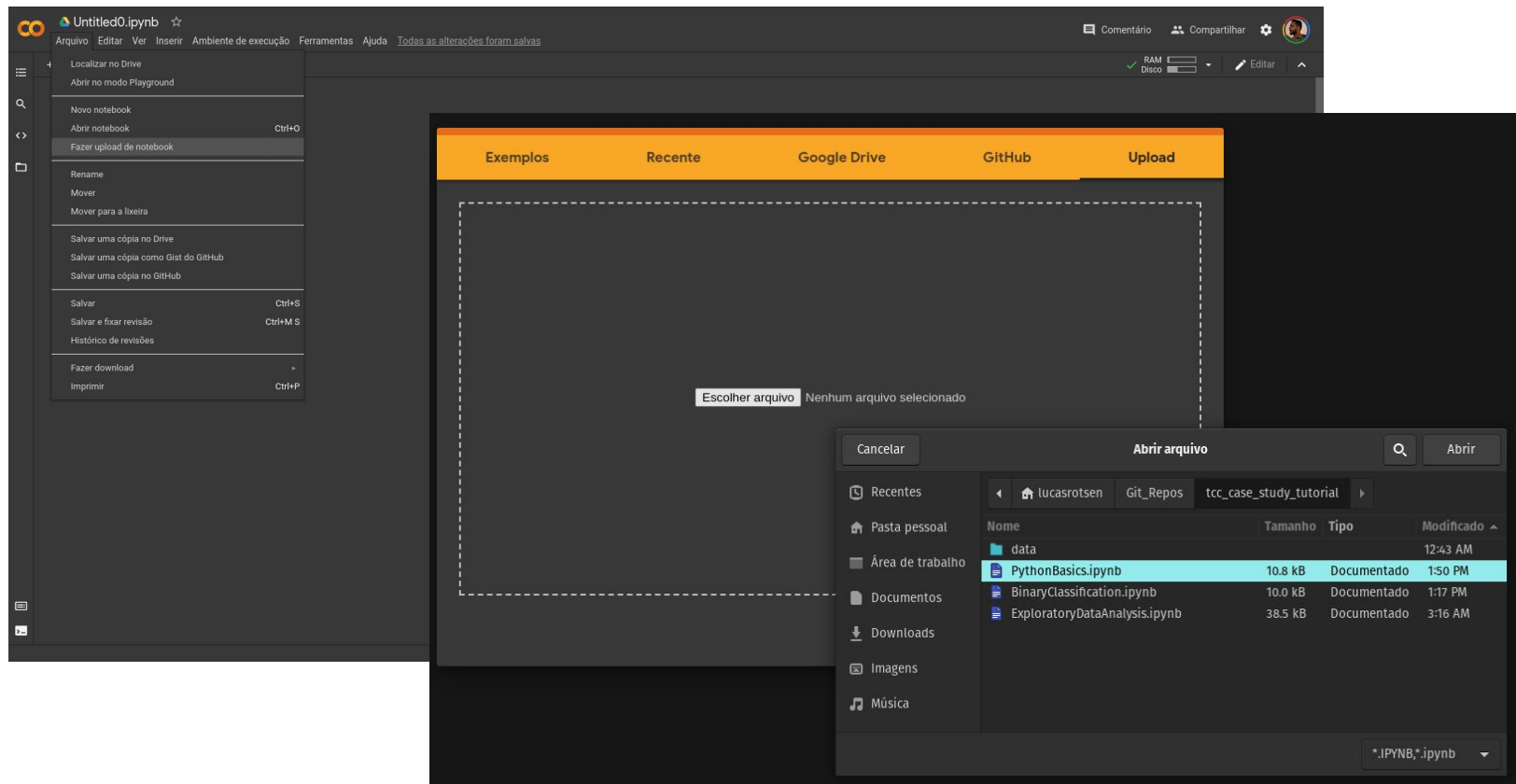
As variáveis definidas em uma célula podem ser usadas mais tarde em outras células:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
    seconds_in_a_week
```

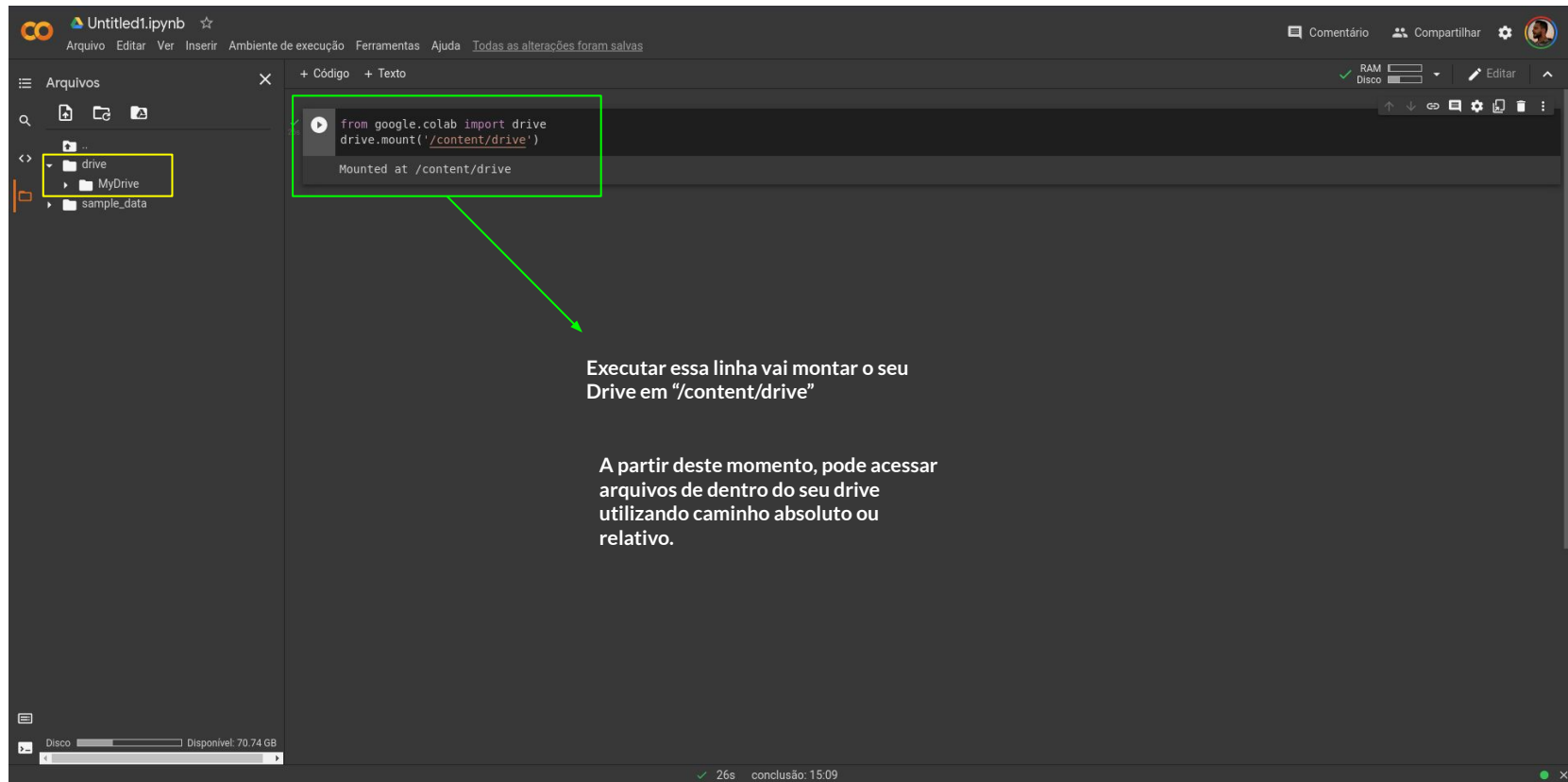
604800

Os notebooks do Colab permitem combinar **código executável** e **rich text** em um só documento, além de **Imagens**, **HTML**, **LaTeX** e muito mais.

# Fazendo o upload de um notebook



# Conectando ao Google Drive



The screenshot shows a Jupyter Notebook environment with a dark theme. On the left, the 'Arquivos' (Files) sidebar is open, showing a file explorer with folders like 'drive', 'MyDrive', and 'sample\_data'. A yellow box highlights the 'drive' folder. In the center, a code cell is selected, containing the following Python code:

```
from google.colab import drive
drive.mount('/content/drive')
```

Below the code, the output shows 'Mounted at /content/drive'. A green box highlights the code cell, and a green arrow points from it to the explanatory text on the right.

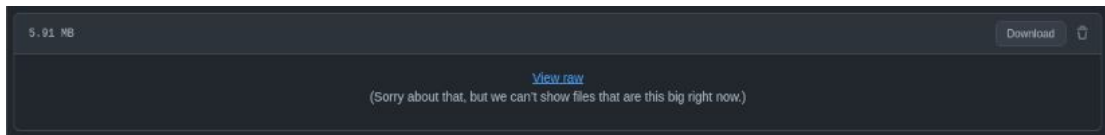
Executar essa linha vai montar o seu Drive em "/content/drive"

A partir deste momento, pode acessar arquivos de dentro do seu drive utilizando caminho absoluto ou relativo.

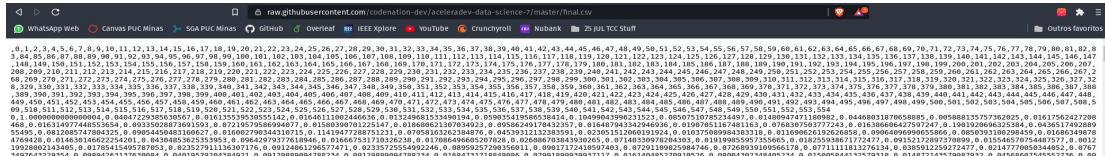
The bottom status bar shows 'Disco' (Disk) usage and 'Disponível: 70.74 GB' (Available: 70.74 GB).

# Acessando o dataset - via URL do GitHub

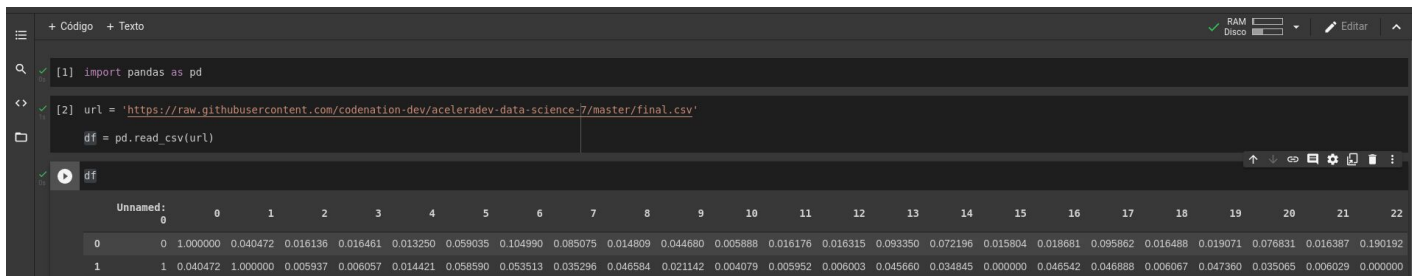
1. Acesse, no GitHub, o(s) arquivo(s) contendo o dataset.
2. Clique para visualizar o arquivo em modo “raw”
3. Copie o link
4. Carregue os dados a partir da URL no pandas



(1)

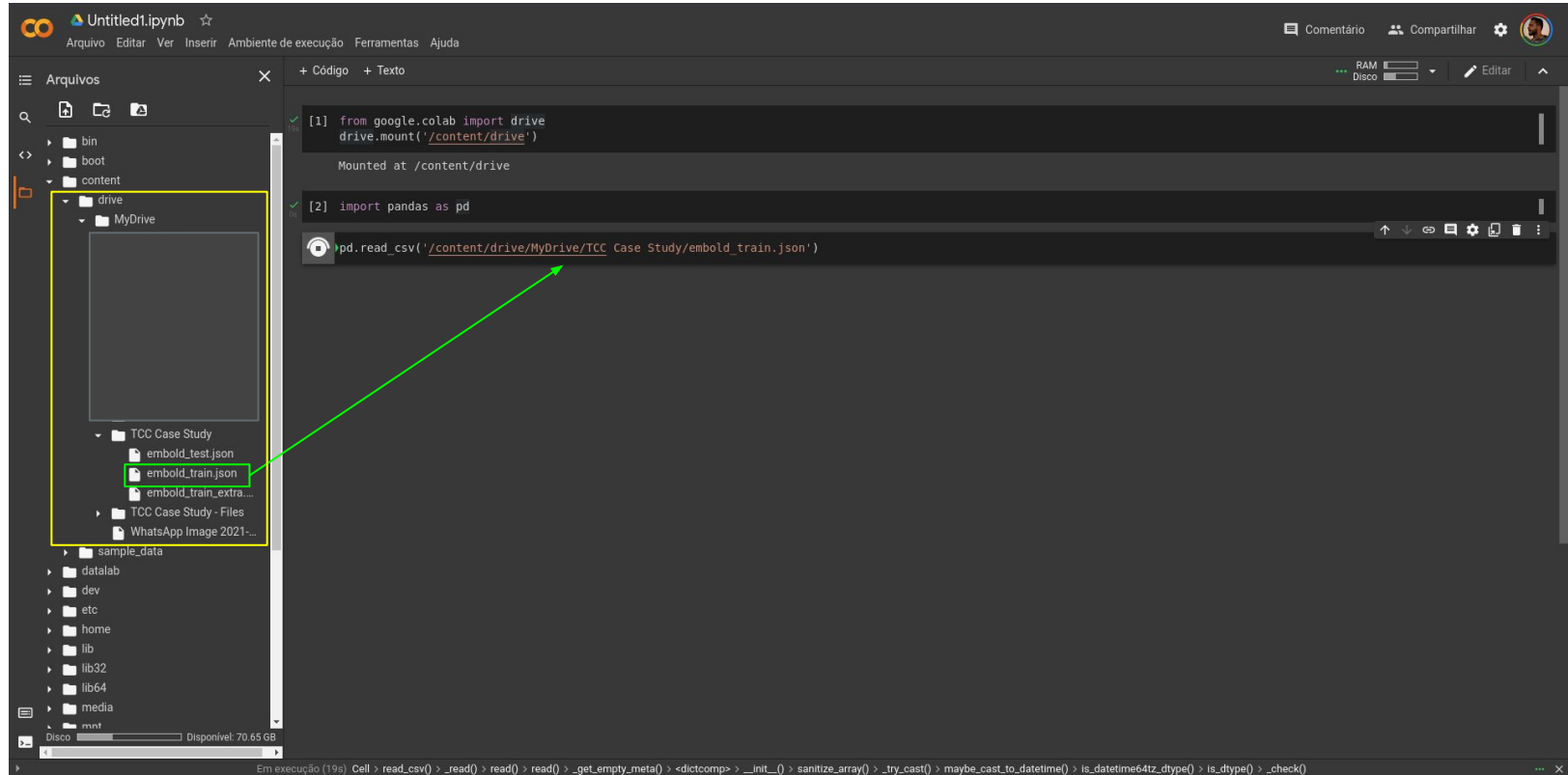


(2)



(3)

# Acessando o dataset - via Google Drive



The screenshot displays a Jupyter Notebook environment with the following components:

- File Explorer (Left Panel):** Shows a directory structure with folders like `bin`, `boot`, `content`, `drive`, `MyDrive`, `TCC Case Study`, `sample_data`, `datalab`, `dev`, `etc`, `home`, `lib`, `lib32`, `lib64`, `media`, and `mnt`. The `MyDrive` folder is expanded, showing subfolders `TCC Case Study` and `TCC Case Study - Files`. Inside `TCC Case Study`, the file `embold_train.json` is highlighted with a green box. A yellow box encompasses the `drive` and `MyDrive` folders.
- Code Editor (Main Area):** Contains two code cells:
  - Cell [1]:

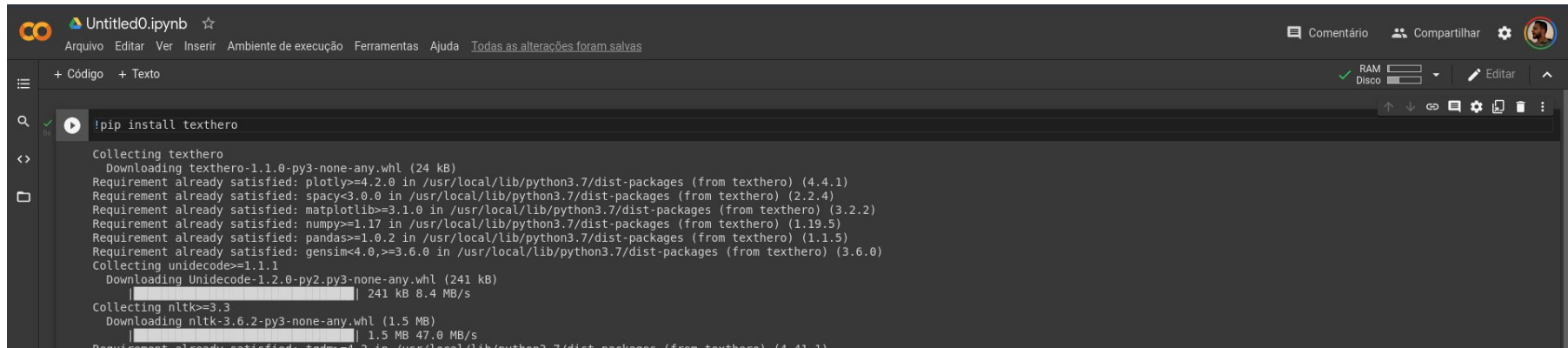
```
from google.colab import drive
drive.mount('/content/drive')
```

Output: `Mounted at /content/drive`
  - Cell [2]:

```
import pandas as pd

pd.read_csv('/content/drive/MyDrive/TCC Case Study/embold_train.json')
```
- Output Area (Right Panel):** Shows the execution progress of the code in cell [2], with a green arrow pointing from the `embold_train.json` file in the file explorer to the file path in the code.
- Status Bar (Bottom):** Displays the current execution state: `Em execução (19s) Cell > read_csv() > _read() > read() > _get_empty_meta() > <dictcomp> > _init_() > sanitize_array() > _try_cast() > maybe_cast_to_datetime() > is_datetime64tz_dtype() > is_dtype() > _check()`.

# Instalando pacotes



Untitled0.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

+ Código + Texto

RAM Disco

Comentário Compartilhar

```
!pip install texthero
```

```
Collecting texthero
  Downloading texthero-1.1.0-py3-none-any.whl (24 kB)
Requirement already satisfied: plotly>=4.2.0 in /usr/local/lib/python3.7/dist-packages (from texthero) (4.4.1)
Requirement already satisfied: spacy<3.0.0 in /usr/local/lib/python3.7/dist-packages (from texthero) (2.2.4)
Requirement already satisfied: matplotlib>=3.1.0 in /usr/local/lib/python3.7/dist-packages (from texthero) (3.2.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from texthero) (1.19.5)
Requirement already satisfied: pandas>=1.0.2 in /usr/local/lib/python3.7/dist-packages (from texthero) (1.1.5)
Requirement already satisfied: gensim<4.0,>=3.6.0 in /usr/local/lib/python3.7/dist-packages (from texthero) (3.6.0)
Collecting unicode>=1.1.1
  Downloading Unidecode-1.2.0-py2.py3-none-any.whl (241 kB)
    | 241 kB 8.4 MB/s
Collecting nltk>=3.3
  Downloading nltk-3.6.2-py3-none-any.whl (1.5 MB)
    | 1.5 MB 47.0 MB/s
Requirement already satisfied: tqdm>=4.3 in /usr/local/lib/python3.7/dist-packages (from texthero) (4.41.1)
```

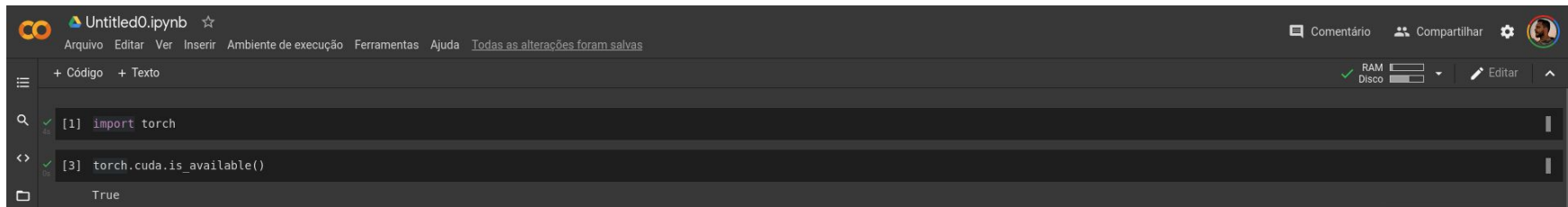


# Habilitando a GPU

The image shows a JupyterLab interface with a notebook titled 'Untitled0.ipynb'. The notebook contains a cell with the command `!pip install textthero`. The output of this cell shows a list of packages being installed, including `textthero`, `plotly`, `spacy`, `matplotlib`, `numpy`, `pandas`, `gensim`, `unicodedata`, `nlTK`, `scikit-learn`, `scipy`, `six`, `smart-open`, `kiwisolver`, `pyarsing`, `cycler`, `python-dateutil`, `joblib`, `click`, `regex`, `pytz`, `retrying`, `srlycl`, `blis`, `preshed`, `catalogue`, `cymem`, `requests`, `setuptools`, `thinc`, `murmurhash`, `wasabi`, `plac`, `importlib-metadata`, `zipp`, `typing-extensions`, and `idna`. The output also shows the download progress for each package.

A dialog box titled 'Configurações de notebook' is open, showing the 'Acelerador de hardware' (Hardware accelerator) section. The 'GPU' option is selected, and the 'TPU' option is also visible. The dialog also shows the 'Omitir saída da célula de código ao salvar este notebook' (Omit code cell output when saving this notebook) checkbox, which is currently unchecked. The 'Cancelar' (Cancel) and 'Salvar' (Save) buttons are at the bottom.

# Testando a GPU



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes the Colab logo, the file name 'Untitled0.ipynb', and a star icon. A menu bar contains 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', 'Ajuda', and a status message 'Todas as alterações foram salvas'. On the right, there are links for 'Comentário', 'Compartilhar', and a user profile icon. Below the top bar, the left sidebar has icons for a menu, search, expand/collapse, and a file explorer. The main area shows two code cells. The first cell contains '[1] import torch' and has a green checkmark icon to its left. The second cell contains '[3] torch.cuda.is\_available()' and also has a green checkmark icon. Below the second cell, the output 'True' is displayed. In the top right corner of the notebook area, there is a status bar showing 'RAM' and 'Disco' usage with progress bars, a dropdown arrow, and an 'Editar' button with an upward arrow.

```
[1] import torch
```

```
[3] torch.cuda.is_available()
```

True