# Tarefa Dois

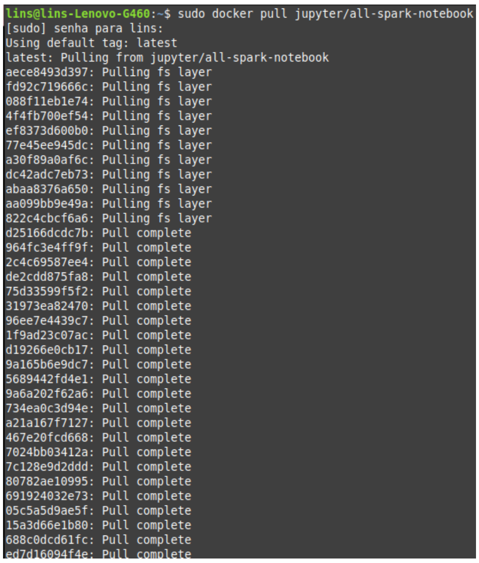
Objetivo: Construir um *job* Spark por meio de um container Docker.

- Nesta atividade faremos uso da imagem **jupyter/all-spark-notebook** (https://registry.hub.docker.com/r/jupyter/all-spark-notebook) para criar um container e utilizar o recurso de shell oferecido pelo Spark. Os passos a executar são:

### 1 - Realizar o *pull* da imagem jupyter/all-spark-notebook

Comando utilizado:

***sudo docker pull jupyter/all-spark-notebook***



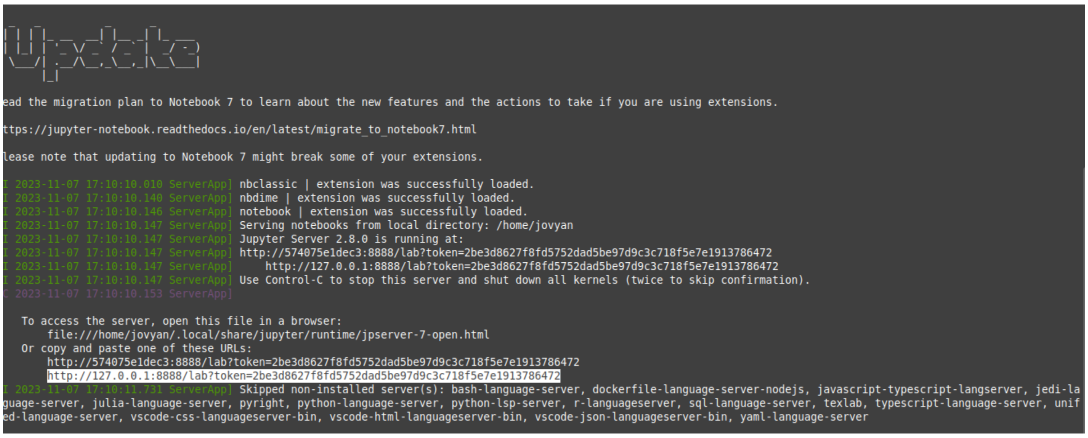
```
lins@lins-Lenovo-G460:~$ sudo docker pull jupyter/all-spark-notebook
[sudo] senha para lins:
Using default tag: latest
latest: Pulling from jupyter/all-spark-notebook
aece8493d397: Pulling fs layer
fd92c719666c: Pulling fs layer
088f11eb1e74: Pulling fs layer
4f4fb700ef54: Pulling fs layer
ef8373d600b0: Pulling fs layer
77e45ee945dc: Pulling fs layer
a30f89a0af6c: Pulling fs layer
dc42adc7eb73: Pulling fs layer
abaa8376a650: Pulling fs layer
aa099bb9e49a: Pulling fs layer
822c4cbcf6a6: Pulling fs layer
d25166dcdc7b: Pull complete
964fc3e4ff9f: Pull complete
2c4c69587ee4: Pull complete
de2cdd875fa8: Pull complete
75d33599f5f2: Pull complete
31973ea82470: Pull complete
96ee7e4439c7: Pull complete
1f9ad23c07ac: Pull complete
d19266e0cb17: Pull complete
9a165b6e9dc7: Pull complete
5689442fd4e1: Pull complete
9a6a202f62a6: Pull complete
734ea0c3d94e: Pull complete
a21a167f7127: Pull complete
467e20fcd668: Pull complete
7024bb03412a: Pull complete
7c128e9d2ddd: Pull complete
80782ae10995: Pull complete
691924032e73: Pull complete
05c5a5d9ae5f: Pull complete
15a3d66e1b80: Pull complete
688c0dcd61fc: Pull complete
ed7d16094f4e: Pull complete
```

Atenção: O tamanho total da imagem é 5.8 GB. Se você não tiver esse espaço disponível, recomendamos utilizar o **Google Colab** para codificar o exercício.

### 2 - Criar um container a partir da imagem

Comando utilizado:

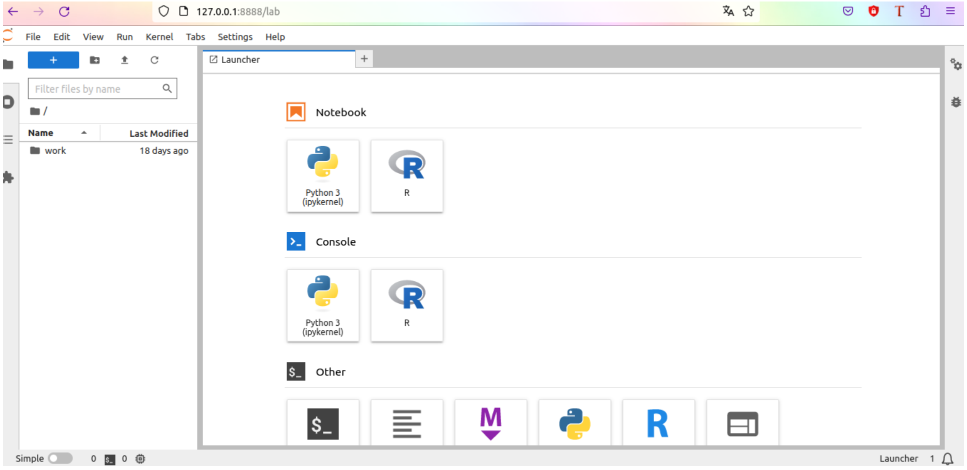
***docker run -it -p 8888:8888 jupyter/all-spark-notebook***



```
 _   _           _       _
| | | |_ __   __| | __ _| |_ ___
| | | | '_ \ / _` |/ _` | __/ _ \
| |_| | |_) | (_| | (_| | ||  __/
 \___/| .__/ \__,_|\__,_|\__\___|
      |_|

ead the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions.

ttps://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html

lease note that updating to Notebook 7 might break some of your extensions.

I 2023-11-07 17:10:10.010 ServerApp] nbclassic | extension was successfully loaded.
I 2023-11-07 17:10:10.140 ServerApp] nbdime | extension was successfully loaded.
I 2023-11-07 17:10:10.146 ServerApp] notebook | extension was successfully loaded.
I 2023-11-07 17:10:10.147 ServerApp] Serving notebooks from local directory: /home/jovyan
I 2023-11-07 17:10:10.147 ServerApp] Jupyter Server 2.8.0 is running at:
I 2023-11-07 17:10:10.147 ServerApp] http://574075e1dec3:8888/lab?token=2be3d8627f8fd5752dad5be97d9c3c718f5e7e1913786472
I 2023-11-07 17:10:10.147 ServerApp]     http://127.0.0.1:8888/lab?token=2be3d8627f8fd5752dad5be97d9c3c718f5e7e1913786472
I 2023-11-07 17:10:10.147 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
C 2023-11-07 17:10:10.153 ServerApp]

   To access the server, open this file in a browser:
       file:///home/jovyan/.local/share/jupyter/runtime/jpserver-7-open.html
   Or copy and paste one of these URLs:
       http://574075e1dec3:8888/lab?token=2be3d8627f8fd5752dad5be97d9c3c718f5e7e1913786472
       http://127.0.0.1:8888/lab?token=2be3d8627f8fd5752dad5be97d9c3c718f5e7e1913786472
I 2023-11-07 17:10:11.731 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langserver, jedi-la
guage-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unif
ed-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```

A url disponibilizada foi:

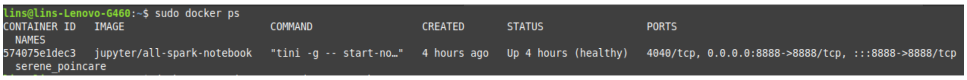
[http://127.0.0.1:8888/lab?token=2be3d8627f8fd5752dad5be97d9c3c718f5e7e1913786472](http://127.0.0.1:8888/lab?token=2be3d8627f8fd5752dad5be97d9c3c718f5e7e1913786472)



**3 - Em outro terminal, execute o comando `pyspark` no seu container. Pesquise sobre o comando *docker exec* para realizar esta ação. Utilize as flags *-i* e *-t* no comando.**

Comando utilizado para buscar o id do container:

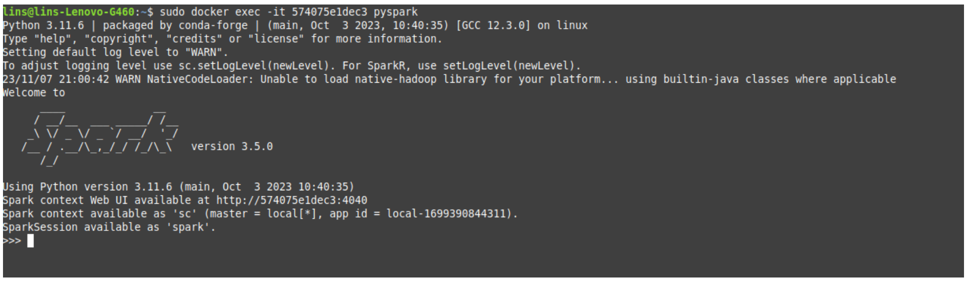
***docker ps***



```
lins@lins-Lenovo-G460:~$ sudo docker ps
CONTAINER ID   IMAGE                        COMMAND                  CREATED       STATUS                 PORTS
  NAMES
574075e1dec3   jupyter/all-spark-notebook   "tini -g -- start-no…"   4 hours ago   Up 4 hours (healthy)   4040/tcp, 0.0.0.0:8888->8888/tcp, :::8888->8888/tcp
  serene_poincare
```

Comando utilizado para executar o pyspark no container:

***sudo docker exec -it 574075e1dec3 pyspark***



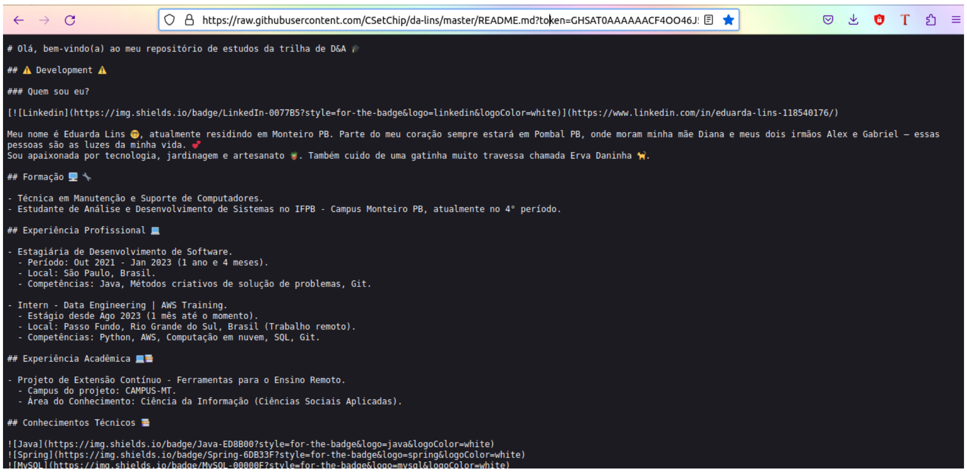
```
lins@lins-Lenovo-G460:~$ sudo docker exec -it 574075e1dec3 pyspark
Python 3.11.6 | packaged by conda-forge | (main, Oct  3 2023, 10:40:35) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/11/07 21:00:42 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/

Using Python version 3.11.6 (main, Oct  3 2023 10:40:35)
Spark context Web UI available at http://574075e1dec3:4040
Spark context available as 'sc' (master = local[*], app id = local-1699390844311).
SparkSession available as 'spark'.
>>> 
```

Dica: Você pode obter arquivos da Internet por meio do comando *wget* no seu container.

- Usando o *Spark Shell,* apresente a sequência de comandos Spark necessários para contar a quantidade de ocorrências de cada palavra contida no arquivo *README.md* de seu repositório *git.*

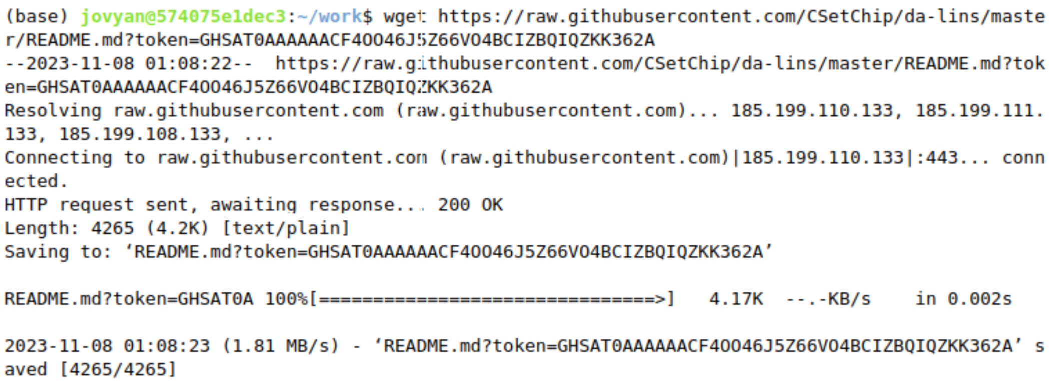
Passos:

Baixei o arquivo README pelo caminho temporário do raw usando o console do JupyterLab:



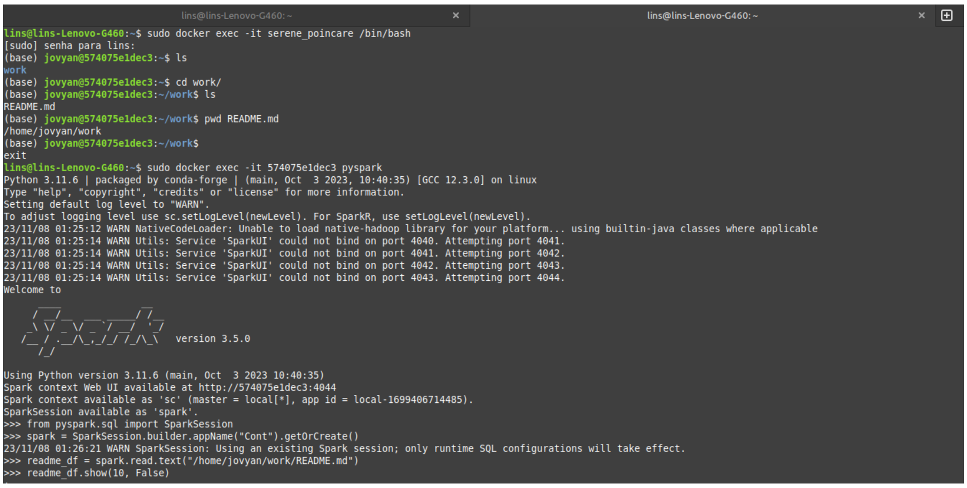
https://raw.githubusercontent.com/CSetChip/da-lins/master/README.md?token=GHSAT0AAAAAACF4OO46J

```
# Olá, bem-vindo(a) ao meu repositório de estudos da trilha de D&A 🎓

## ⚠ Development ⚠

### Quem sou eu?

[![Linkedin](https://img.shields.io/badge/LinkedIn-0077B5?style=for-the-badge&logo=linkedin&logoColor=white)](https://www.linkedin.com/in/eduarda-lins-118540176/)

Meu nome é Eduarda Lins 😊, atualmente residindo em Monteiro PB. Parte do meu coração sempre estará em Pombal PB, onde moram minha mãe Diana e meus dois irmãos Alex e Gabriel — essas
pessoas são as luzes da minha vida. 💕
Sou apaixonada por tecnologia, jardinagem e artesanato 🪴. Também cuido de uma gatinha muito travessa chamada Erva Daninha 🐈.

## Formação 💻 🔧

- Técnica em Manutenção e Suporte de Computadores.
- Estudante de Análise e Desenvolvimento de Sistemas no IFPB - Campus Monteiro PB, atualmente no 4° período.

## Experiência Profissional 💻

- Estagiária de Desenvolvimento de Software.
  - Período: Out 2021 - Jan 2023 (1 ano e 4 meses).
  - Local: São Paulo, Brasil.
  - Competências: Java, Métodos criativos de solução de problemas, Git.

- Intern - Data Engineering | AWS Training.
  - Estágio desde Ago 2023 (1 mês até o momento).
  - Local: Passo Fundo, Rio Grande do Sul, Brasil (Trabalho remoto).
  - Competências: Python, AWS, Computação em nuvem, SQL, Git.

## Experiência Acadêmica 💻📚

- Projeto de Extensão Contínuo - Ferramentas para o Ensino Remoto.
  - Campus do projeto: CAMPUS-MT.
  - Área do Conhecimento: Ciência da Informação (Ciências Sociais Aplicadas).

## Conhecimentos Técnicos 📚

![Java](https://img.shields.io/badge/Java-ED8B00?style=for-the-badge&logo=java&logoColor=white)
![Spring](https://img.shields.io/badge/Spring-6DB33F?style=for-the-badge&logo=spring&logoColor=white)
![MySQL](https://img.shields.io/badge/MySQL-00000F?style=for-the-badge&logo=mysql&logoColor=white)
```



```
(base) jovyan@574075e1dec3:~/work$ wget https://raw.githubusercontent.com/CSetChip/da-lins/maste
r/README.md?token=GHSAT0AAAAAACF40046J5Z66VO4BCIZBQIQZKK362A
--2023-11-08 01:08:22--  https://raw.githubusercontent.com/CSetChip/da-lins/master/README.md?tok
en=GHSAT0AAAAAACF40046J5Z66VO4BCIZBQIQZKK362A
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.
133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... conn
ected.
HTTP request sent, awaiting response... 200 OK
Length: 4265 (4.2K) [text/plain]
Saving to: ‘README.md?token=GHSAT0AAAAAACF40046J5Z66VO4BCIZBQIQZKK362A’

README.md?token=GHSAT0A 100%[===============================>]   4.17K  --.-KB/s    in 0.002s

2023-11-08 01:08:23 (1.81 MB/s) - ‘README.md?token=GHSAT0AAAAAACF40046J5Z66VO4BCIZBQIQZKK362A’ s
aved [4265/4265]
```

Converti o arquivos para README.md, em seguida voltei para o terminal para verificar o caminho:



```
lins@lins-Lenovo-G460:~$ sudo docker exec -it serene_poincare /bin/bash
[sudo] senha para lins:
(base) jovyan@574075e1dec3:~$ ls
work
(base) jovyan@574075e1dec3:~$ cd work/
(base) jovyan@574075e1dec3:~/work$ ls
README.md
(base) jovyan@574075e1dec3:~/work$ pwd README.md
/home/jovyan/work
(base) jovyan@574075e1dec3:~/work$
exit
lins@lins-Lenovo-G460:~$ sudo docker exec -it 574075e1dec3 pyspark
Python 3.11.6 | packaged by conda-forge | (main, Oct  3 2023, 10:40:35) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/11/08 01:25:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
23/11/08 01:25:14 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
23/11/08 01:25:14 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
23/11/08 01:25:14 WARN Utils: Service 'SparkUI' could not bind on port 4042. Attempting port 4043.
23/11/08 01:25:14 WARN Utils: Service 'SparkUI' could not bind on port 4043. Attempting port 4044.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/

Using Python version 3.11.6 (main, Oct  3 2023 10:40:35)
Spark context Web UI available at http://574075e1dec3:4044
Spark context available as 'sc' (master = local[*], app id = local-1699406714485).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession.builder.appName("Cont").getOrCreate()
23/11/08 01:26:21 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>> readme_df = spark.read.text("/home/jovyan/work/README.md")
>>> readme_df.show(10, False)
```

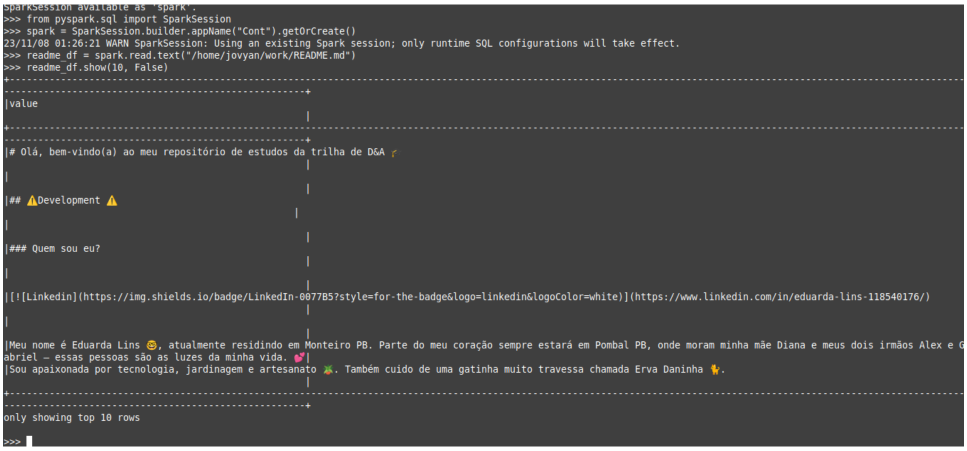
Utilizei os seguintes comandos para carregar o arquivo no pyspark e printar na tela:

***>>> from pyspark.sql import SparkSession***

***>>> spark = SparkSession.builder.appName("Cont").getOrCreate()***

***>>> readme_df = spark.read.text("/home/jovyan/work/README.md")***

***>>> readme_df.show(10, False)***



```
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession.builder.appName("Cont").getOrCreate()
23/11/08 01:26:21 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>> readme_df = spark.read.text("/home/jovyan/work/README.md")
>>> readme_df.show(10, False)
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
|value
                                                     |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
|# Olá, bem-vindo(a) ao meu repositório de estudos da trilha de D&A 🎓
                                                     |
|
                                                     |
|## ⚠️Development ⚠️
                                                   |
|
                                                     |
|### Quem sou eu?
                                                     |
|
                                                     |
|[![Linkedin](https://img.shields.io/badge/LinkedIn-0077B5?style=for-the-badge&logo=linkedin&logoColor=white)](https://www.linkedin.com/in/eduarda-lins-118540176/)
                                                     |
|
                                                     |
|Meu nome é Eduarda Lins 🤓, atualmente residindo em Monteiro PB. Parte do meu coração sempre estará em Pombal PB, onde moram minha mãe Diana e meus dois irmãos Alex e G
abriel — essas pessoas são as luzes da minha vida. 💕|
|Sou apaixonada por tecnologia, jardinagem e artesanato 🪴. Também cuido de uma gatinha muito travessa chamada Erva Daninha 🐈.
                                                     |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------+
only showing top 10 rows

>>> █
```

Os seguintes comandos foram utilizados para realizar a contagem e mostrar as 10 primeiras linhas:

***>>> from pyspark.sql.functions import explode, split, lower***

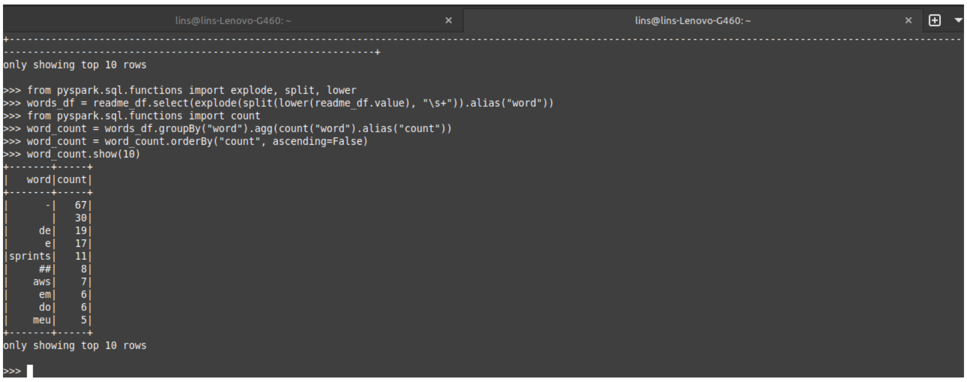
***>>> words_df = readme_df.select(explode(split(lower(readme_df.value), "\s+")).alias("word"))***

***>>> from pyspark.sql.functions import count***

***>>>word_count=words_df.groupBy("word").agg(count("word").alias("co***
***u nt"))***

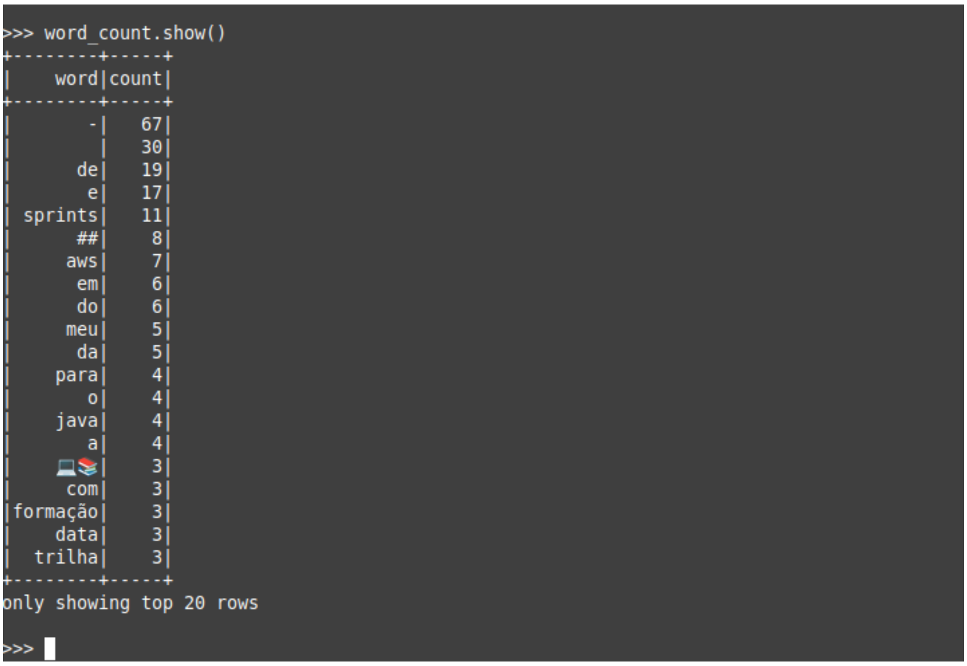
***>>> word_count = word_count.orderBy("count", ascending=False)***

***>>> word_count.show(10)***



```
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------+
only showing top 10 rows

>>> from pyspark.sql.functions import explode, split, lower
>>> words_df = readme_df.select(explode(split(lower(readme_df.value), "\s+")).alias("word"))
>>> from pyspark.sql.functions import count
>>> word_count = words_df.groupBy("word").agg(count("word").alias("count"))
>>> word_count = word_count.orderBy("count", ascending=False)
>>> word_count.show(10)
+-------+-----+
|   word|count|
+-------+-----+
|      -|   67|
|       |   30|
|     de|   19|
|      e|   17|
|sprints|   11|
|     ##|    8|
|    aws|    7|
|     em|    6|
|     do|    6|
|    meu|    5|
+-------+-----+
only showing top 10 rows

>>> 
```

Por último ultilisei o “.show()” semparametro para listar todas as palavras e contagens:



```
>>> word_count.show()
+--------+-----+
|    word|count|
+--------+-----+
|       -|   67|
|        |   30|
|      de|   19|
|       e|   17|
| sprints|   11|
|      ##|    8|
|     aws|    7|
|      em|    6|
|      do|    6|
|     meu|    5|
|      da|    5|
|    para|    4|
|       o|    4|
|    java|    4|
|       a|    4|
|    💻📚|    3|
|     com|    3|
|formação|    3|
|    data|    3|
|  trilha|    3|
+--------+-----+
only showing top 20 rows

>>> 
```