```
!pip install duckdb pandas
import duckdb
import pandas as pd
#importei as bibilhotecas
```

```
Requirement already satisfied: duckdb in /usr/local/lib/python3.12/dist-packages (1.3.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0
```

```
from google.colab import files
uploaded = files.upload()
#uppei o arquivo disponibilizado
```

Escolher Ficheiros 2 ficheiros
googleplaystore_user_reviews.csv(text/csv) - 7669276 bytes, last modified: 16/10/2025 - 100% done
googleplaystore.csv(text/csv) - 1360155 bytes, last modified: 16/10/2025 - 100% done
Saving googleplaystore_user_reviews.csv to googleplaystore_user_reviews.csv
Saving googleplaystore.csv to googleplaystore.csv

```
apps = pd.read_csv('googleplaystore.csv')
reviews = pd.read_csv('googleplaystore_user_reviews.csv')

# Conectei ao DuckDB
conn = duckdb.connect(database=':memory:')

# Registrei tabelas
conn.register('apps_raw', apps)
conn.register('reviews_raw', reviews)
```

```
<duckdb.duckdb.DuckDBPyConnection at 0x7c750d1a2b30>
```

```
query_apps = """
WITH base AS (
    SELECT
        upper(substr(Category,1,1)) || lower(substr(Category,2)) AS Category,
        TRIM(App) AS App,
        COALESCE(TRY_CAST(NULLIF(Rating, 'NaN') AS DOUBLE),
                 AVG(TRY_CAST(NULLIF(Rating, 'NaN') AS DOUBLE))
                 OVER (PARTITION BY Category)) AS Rating_Imputed,
        TRY_CAST(Reviews AS BIGINT) AS Reviews,
        TRY_CAST(REPLACE(REPLACE(REPLACE(Installs, '+', ''), ',', ''), 'Free', '0') AS BIGINT) AS Installs,
        Type,
        TRY_CAST(REPLACE(Price, '$', '') AS DOUBLE) AS Price,
        "Content Rating" AS Content_Rating,
        Genres,
        CASE
            WHEN "Last Updated" LIKE '%January%' OR "Last Updated" LIKE '%February%' OR
                 "Last Updated" LIKE '%March%' OR "Last Updated" LIKE '%April%' OR
                 "Last Updated" LIKE '%May%' OR "Last Updated" LIKE '%June%' OR
                 "Last Updated" LIKE '%July%' OR "Last Updated" LIKE '%August%' OR
                 "Last Updated" LIKE '%September%' OR "Last Updated" LIKE '%October%' OR
                 "Last Updated" LIKE '%November%' OR "Last Updated" LIKE '%December%'
            THEN STRPTIME("Last Updated", '%B %d, %Y')
            ELSE NULL
        END AS Last_Updated
    FROM apps_raw
)
SELECT
    *,
    EXTRACT(YEAR FROM Last_Updated) AS Ano_Atualizacao,
    CASE
        WHEN Installs < 100000 THEN 'Baixo'
        WHEN Installs BETWEEN 100000 AND 1000000 THEN 'Médio'
        ELSE 'Alto'
    END AS Reach_Band,
    CASE WHEN Type = 'Paid' THEN Installs * Price ELSE 0 END AS Revenue_Potential
FROM base
"""
apps_clean = conn.execute(query_apps).df()
apps_clean.head(10)
```

	Category	App	Rating_Imputed	Reviews	Installs	Type	Price	Content_Rating	Genres	Last_Updated	Ano_Atualizacao	Rea
0	Beauty	Beard Live Camera Photo Editor	4.7	900	5000	Free	0.0	Everyone	Beauty	2018-03-25	2018	
1	Beauty	Hush - Beauty for Everyone	4.7	18900	500000	Free	0.0	Everyone	Beauty	2018-08-02	2018	
2	Beauty	Female Daily	3.9	4354	100000	Free	0.0	Teen	Beauty	2018-08-06	2018	
3	Beauty	Skin Disease	4.0	1	100	Free	0.0	Everyone	Beauty	2017-08-30	2017	
		Mirror -										

```
query_reviews = """
SELECT
    TRIM(App) AS App,
    Translated_Review,
    upper(substr(Sentiment,1,1)) || lower(substr(Sentiment,2)) AS Sentiment,
    round(Sentiment_Polarity,2) as Sentiment_Polarity,
    round(Sentiment_Subjectivity,2) as Sentiment_Subjectivity
FROM reviews_raw
WHERE Translated_Review IS NOT NULL
"""
reviews_clean = conn.execute(query_reviews).df()
reviews_clean.head(10)
```

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.53
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.29
2	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.88
3	10 Best Foods for You	Best idea us	Positive	1.00	0.30
4	10 Best Foods for You	Best way	Positive	1.00	0.30
5	10 Best Foods for You	Amazing	Positive	0.60	0.90
6	10 Best Foods for You	Looking forward app,	Neutral	0.00	0.00
7	10 Best Foods for You	It helpful site ! It help foods get !	Neutral	0.00	0.00
8	10 Best Foods for You	good you.	Positive	0.70	0.60
9	10 Best Foods for You	Useful information The amount spelling errors ...	Positive	0.20	0.10

Passos seguintes: Gerar código com reviews_clean | New interactive sheet

```
apps_clean.to_csv("apps_tratados.csv", index=False, sep=";", decimal=",")
reviews_clean.to_csv("reviews_tratados.csv", index=False, sep=";", decimal=",")
from google.colab import files
files.download("apps_tratados.csv")
files.download("reviews_tratados.csv")
```