Criando Aplicativos web som streamlit

url: <https://hub.asimov.academy/curso/atividade/introducao-ao-streamlit/>

git: <https://github.com/luizfernandodag/ASIMOV-https-hub.asimov.academy-curso-atividade-introducao-ao-streamlit>

drive: <https://drive.google.com/file/d/10ODXxd2ug9uF1WrtrOFGKn7Id0Sdut3Q/view>

Sumário

[Introdução ao Streamlit 1](#_Toc192952557)

[Introdução ao Streamlit 1](#_Toc192952558)

[Rodando o primeiro Dashboard 1](#_Toc192952559)

[Apresentando dados no Streamlit 1](#_Toc192952560)

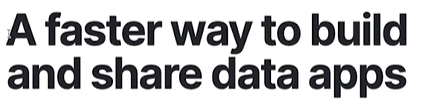
[Quiz – Introdução ao Streamlit 1](#_Toc192952561)

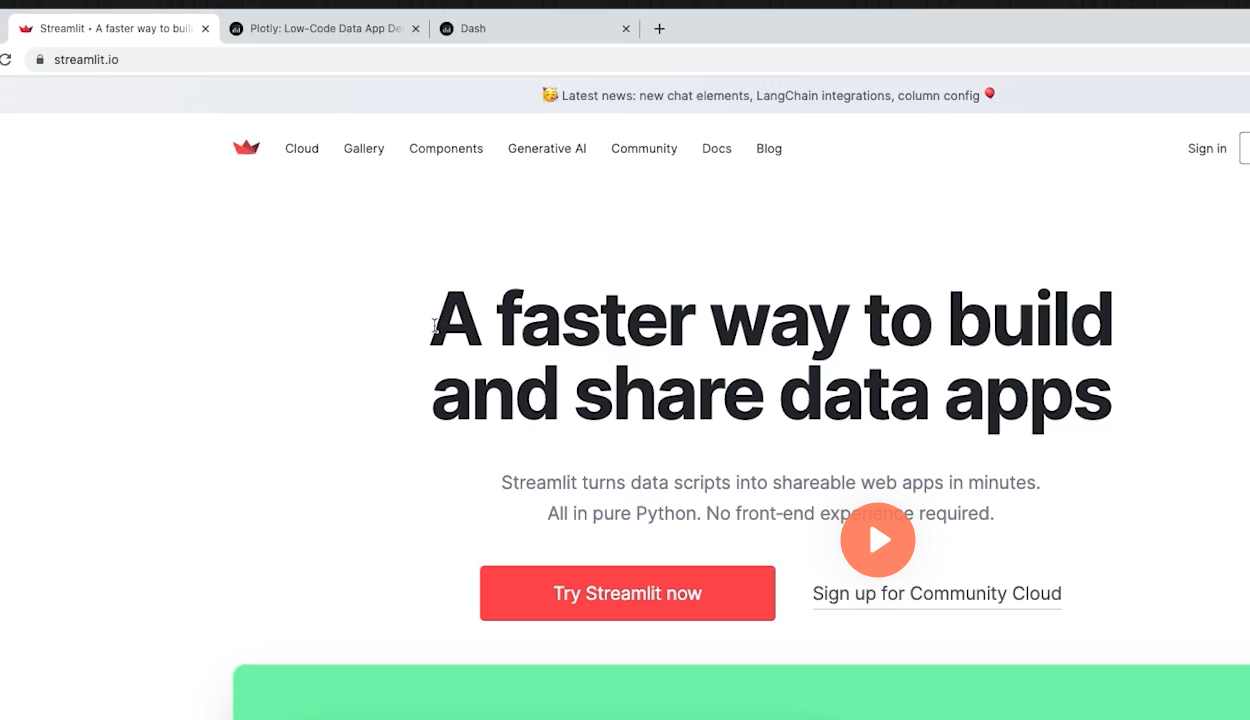
# [Introdução ao Streamlit](https://hub.asimov.academy/curso/atividade/introducao-ao-streamlit/)

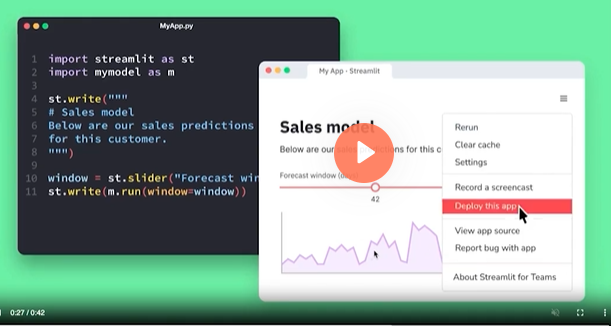
## [Introdução ao Streamlit](https://hub.asimov.academy/curso/atividade/introducao-ao-streamlit/)

Streamlit X plotli -> aplicativos web

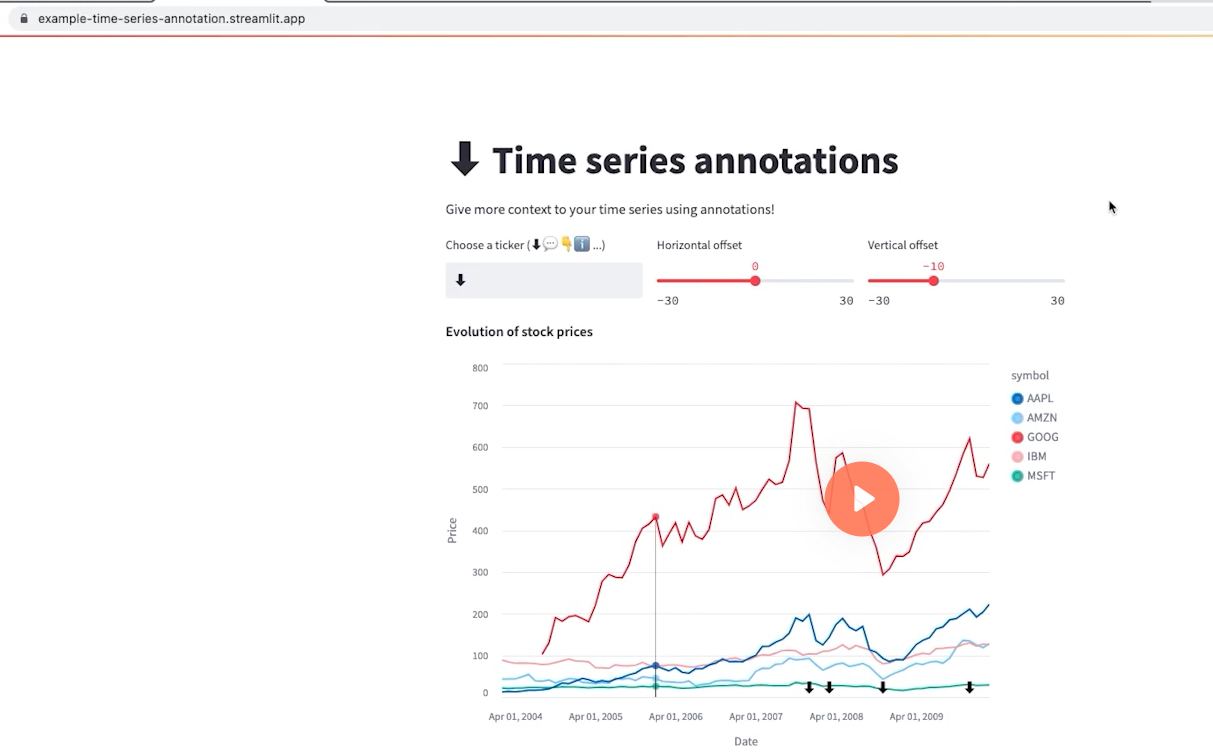
<https://streamlit.io/>







### Exemplos de Projetos





## [Rodando o primeiro Dashboard](https://hub.asimov.academy/curso/atividade/rodando-o-primeiro-dashboard/)

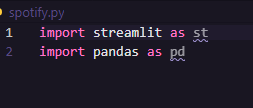
**pip install streamlit**

****

-Começando o código

**import streamlit as st**

**import pandas as pd**



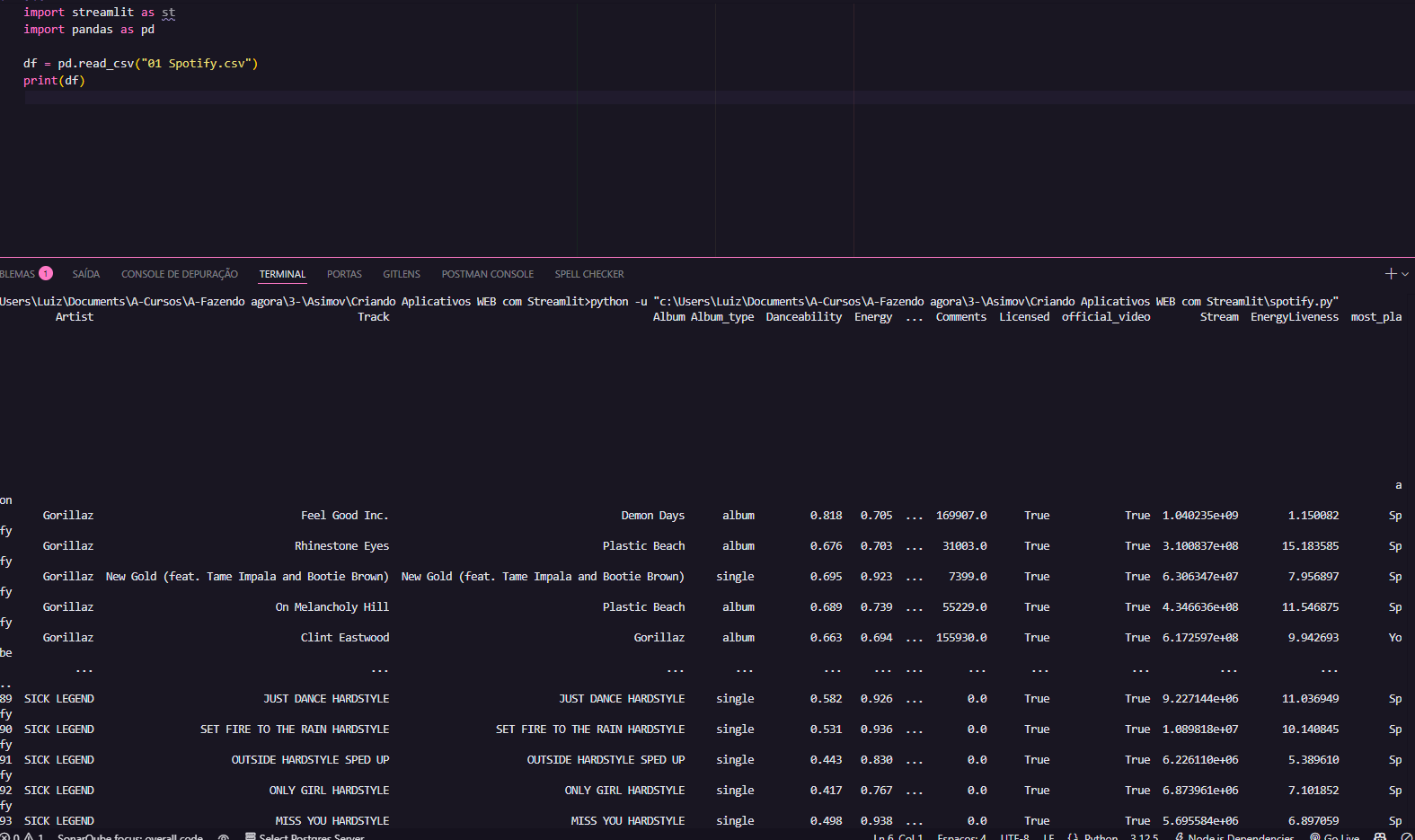
- vamos ler o a\rquivo csv

**import streamlit as st**

**import pandas as pd**

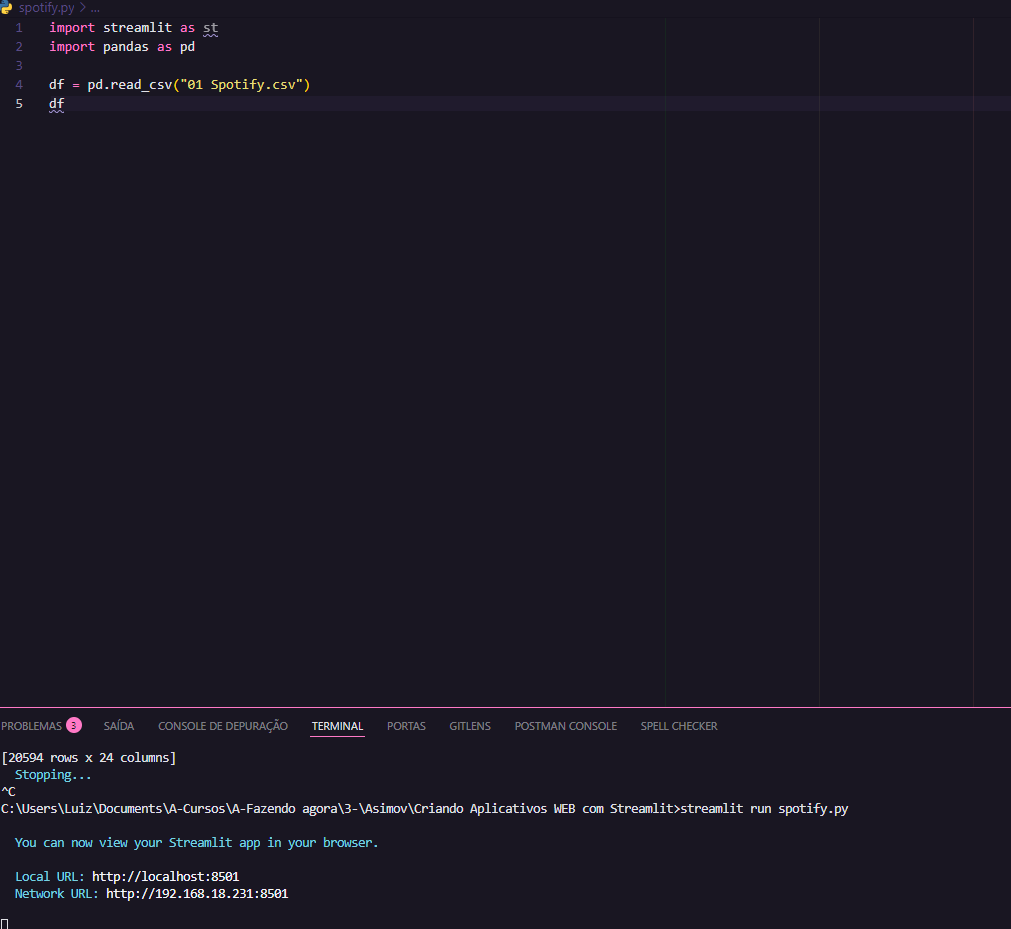
**df = pd.read\_csv("01 Spotify.csv")**

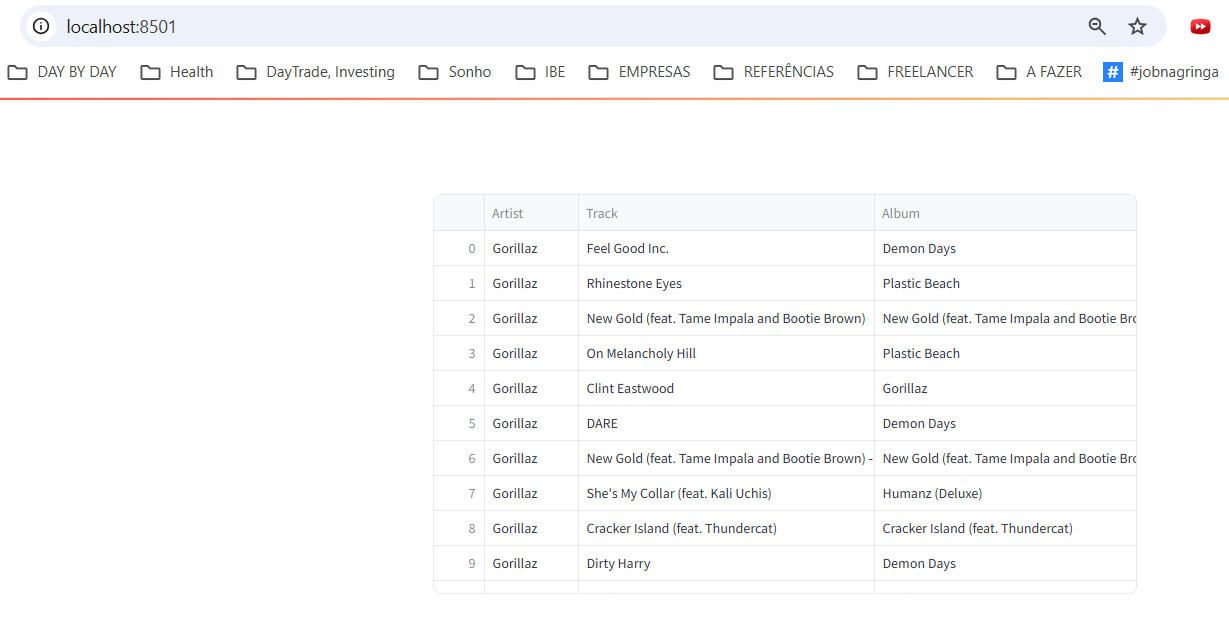
**df**



- vamos rodar o **spotify.py**  no streamlit

**streamlit run spotify.py**

****

****

## [Apresentando dados no Streamlit](https://hub.asimov.academy/curso/atividade/apresentando-dados-no-streamlit/)

Stream lit é um liveserver

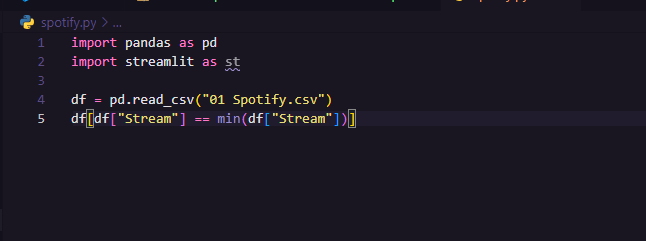
Exemplo: vamos mostrar apenas os que o mínimo de Stream:

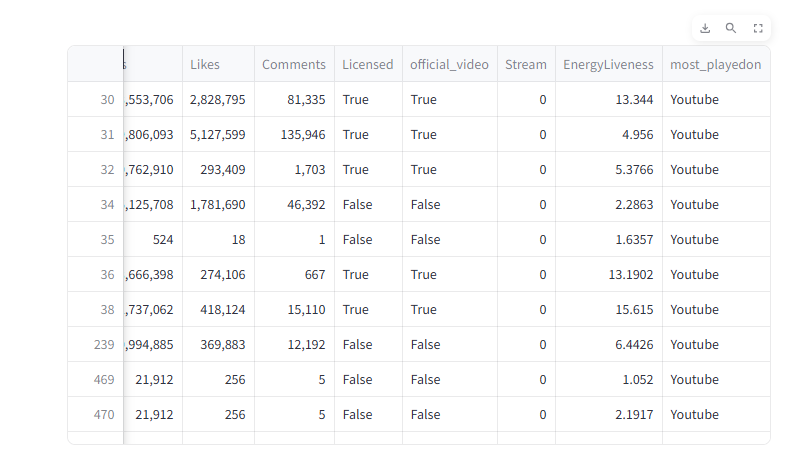
**import pandas as pd**

**import streamlit as st**

**df = pd.read\_csv("01 Spotify.csv")**

**df[df["Stream"] == min(df["Stream"])]**

****

****

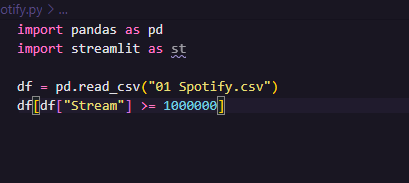
Exemplo: vamos mostrar apenas as músicas com ais de 1 milhão de views

**import pandas as pd**

**import streamlit as st**

**df = pd.read\_csv("01 Spotify.csv")**

**df[df["Stream"] >= 1000000]**

****



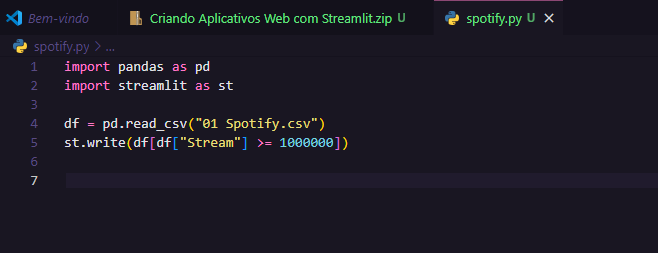
Nesse caso o **st.write()**  está implicíto

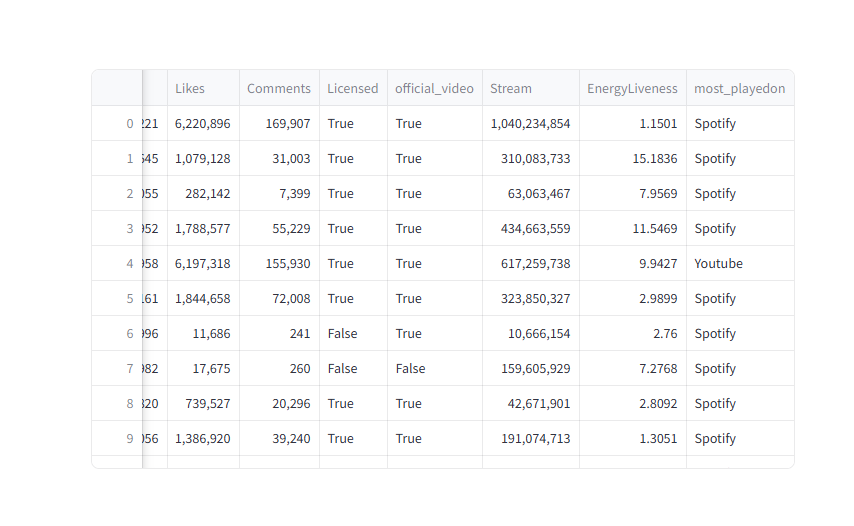
**import pandas as pd**

**import streamlit as st**

**df = pd.read\_csv("01 Spotify.csv")**

**st.write(df[df["Stream"] >= 1000000])**

****

****

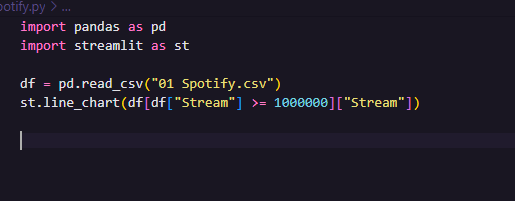
**--**Vamos montar um gráfico bem simples da coluna **Stream**

**import pandas as pd**

**import streamlit as st**

**df = pd.read\_csv("01 Spotify.csv")**

**st.line\_chart(df[df["Stream"] >= 1000000]["Stream"])**





- vamos setar o index como o nome do artista

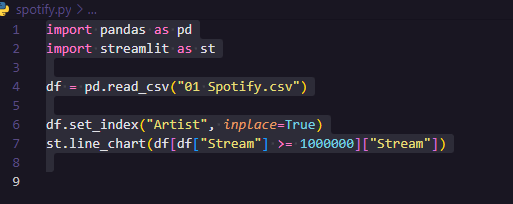
**import pandas as pd**

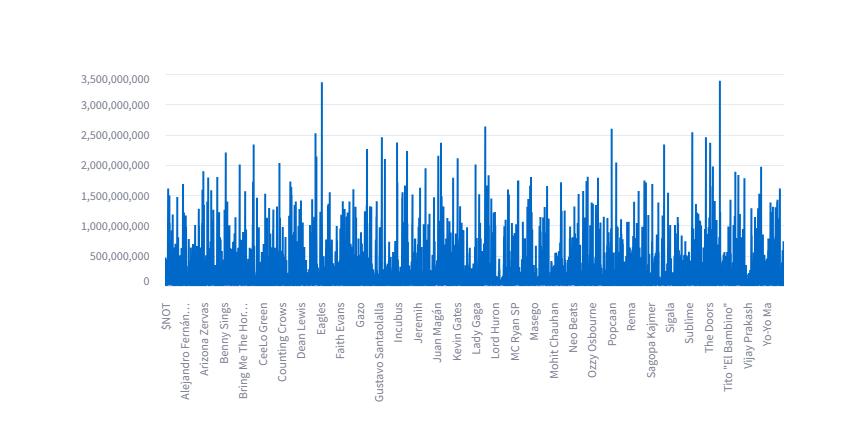
**import streamlit as st**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Artist", *inplace*=True)**

**st.line\_chart(df[df["Stream"] >= 1000000]["Stream"])**

****

****

-A construção do dataframe sempre se dá de cima para baixp

**import pandas as pd**

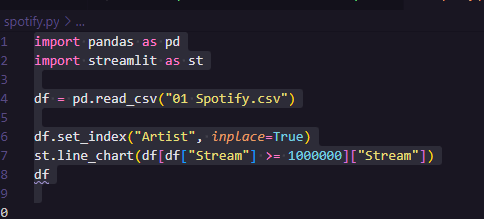
**import streamlit as st**

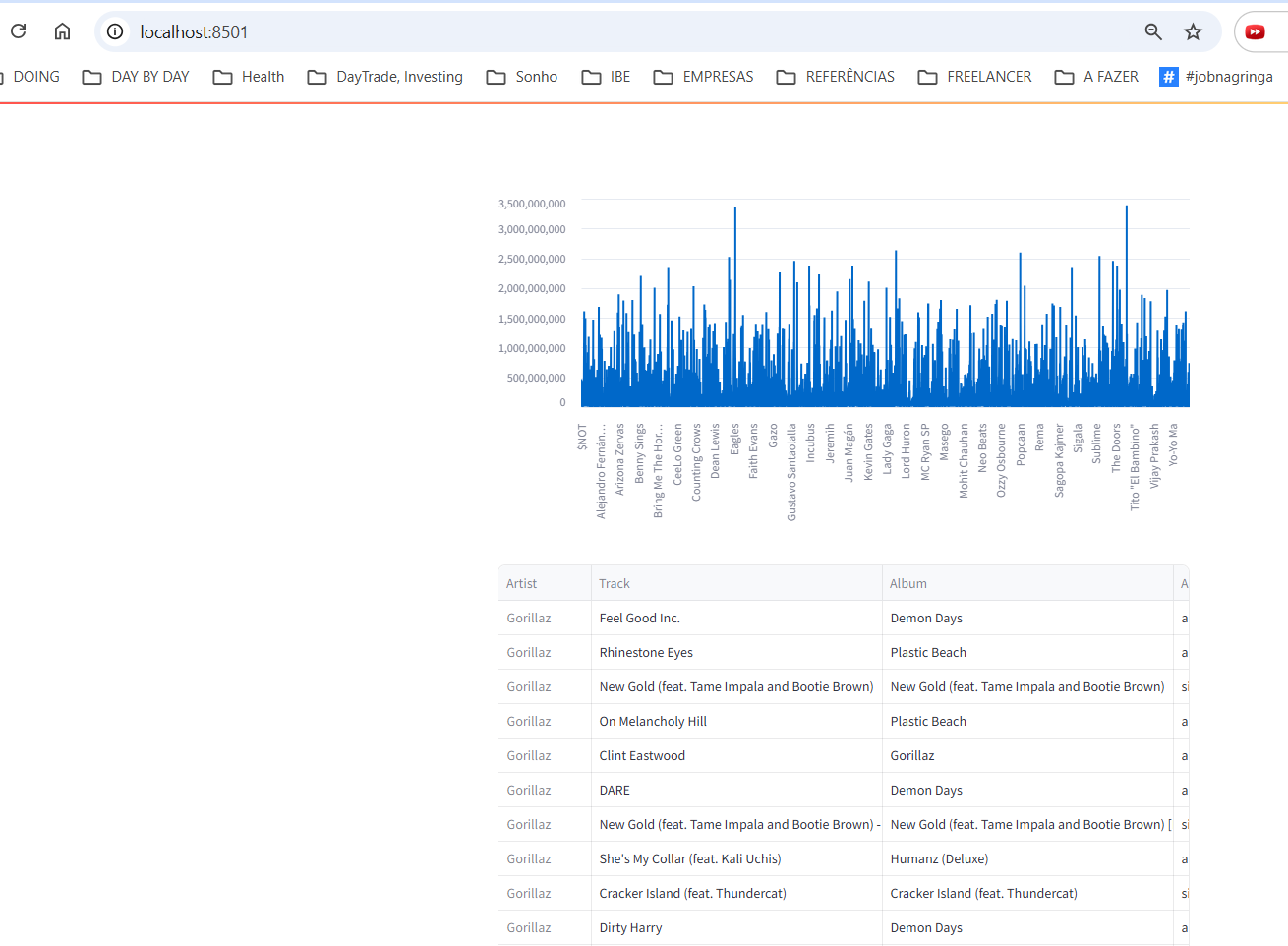
**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Artist", *inplace*=True)**

**st.line\_chart(df[df["Stream"] >= 1000000]["Stream"])**

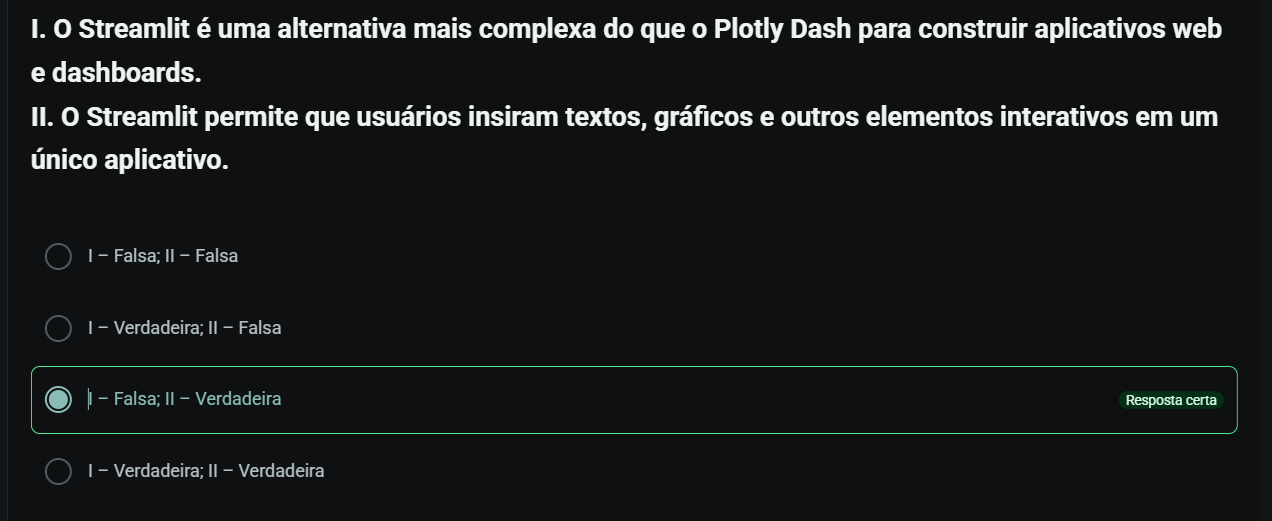
**df**

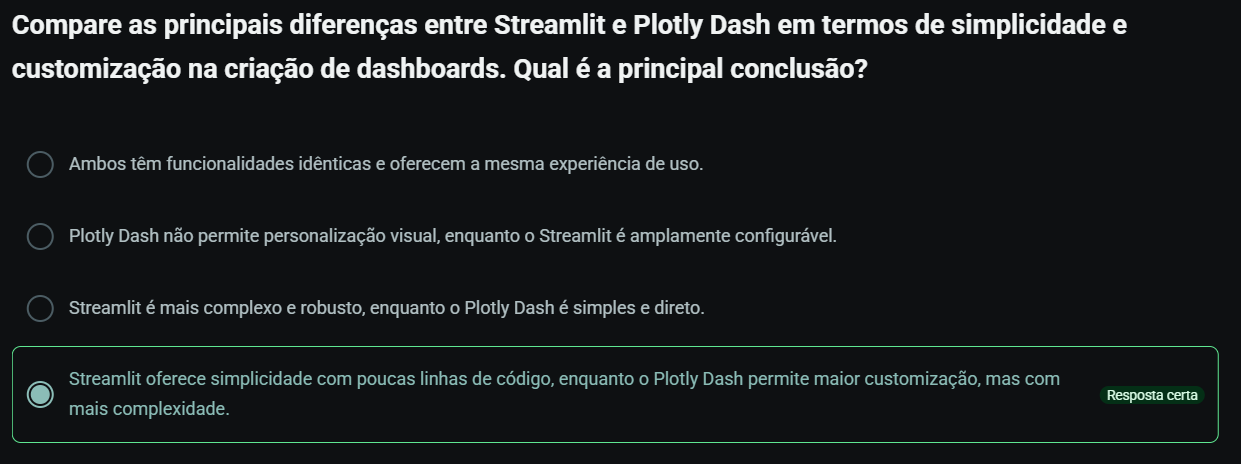
****

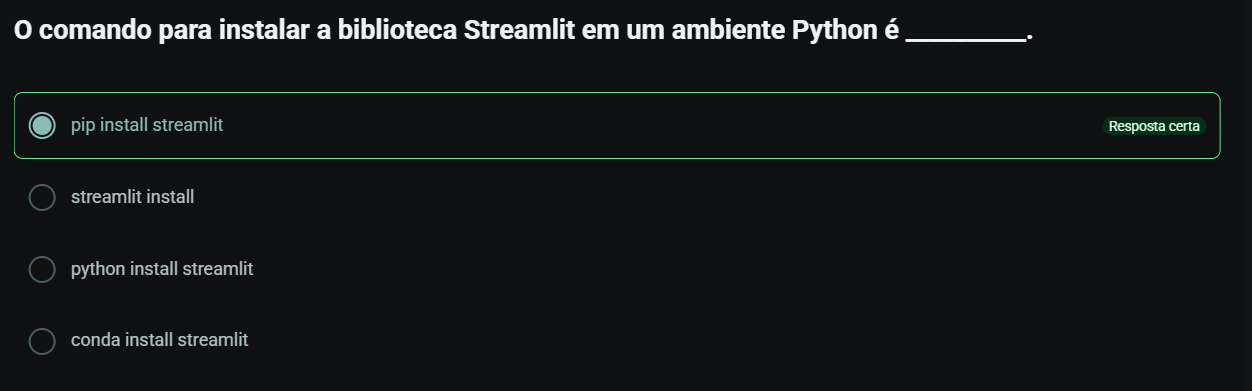
****

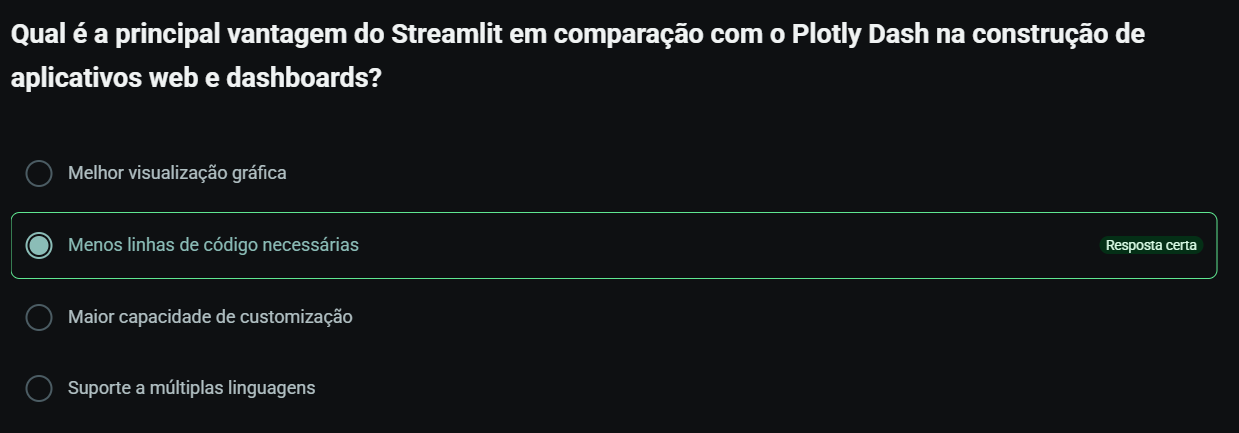
## [Quiz – Introdução ao Streamlit](https://hub.asimov.academy/curso/atividade/quiz-introducao-ao-streamlit/)











# Como customizar seu aplicativo

## [Input Widgets](https://hub.asimov.academy/curso/atividade/input-widgets/)

Para filtrar os dados de alguma forma



Vamos selecionar as músicas com mais de um bilhão de Streams:

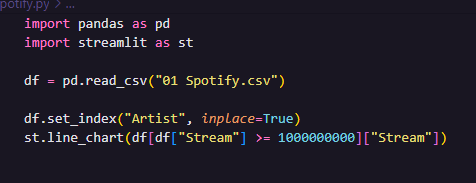
**import pandas as pd**

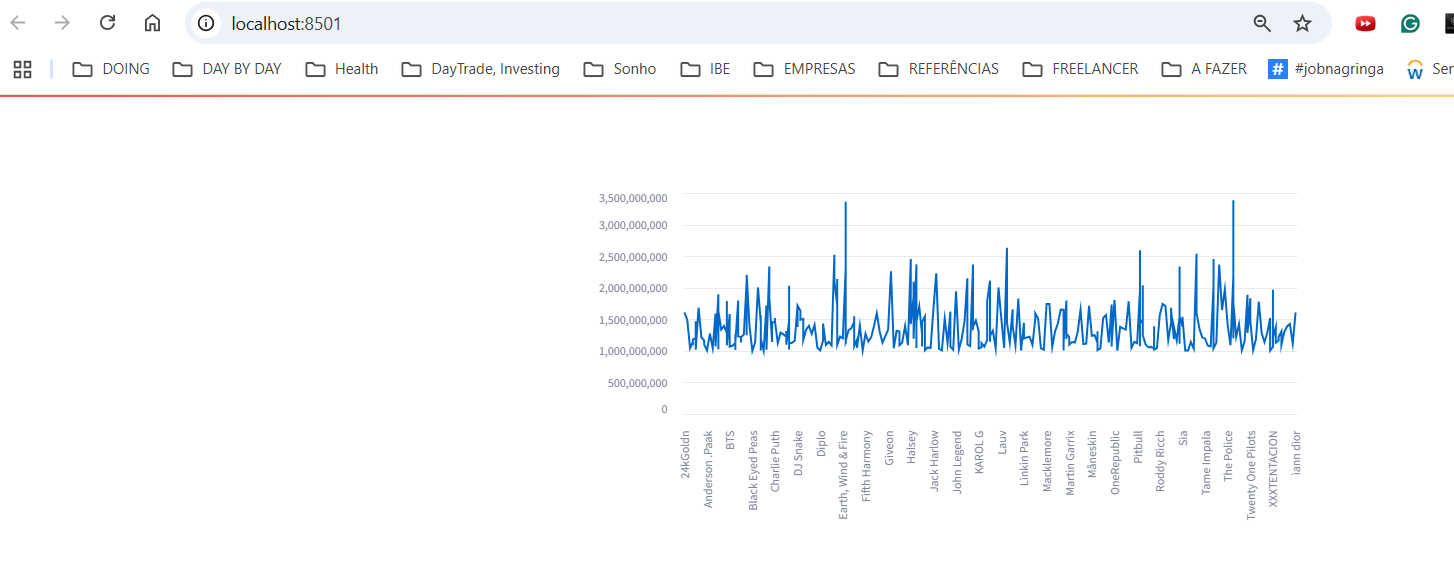
**import streamlit as st**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Artist", *inplace*=True)**

**st.line\_chart(df[df["Stream"] >= 1000000000]["Stream"])**





- Vamos usar um gráfico de barras, setar as configurações da página para einde e colocar o título e usar **Track**  como index

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

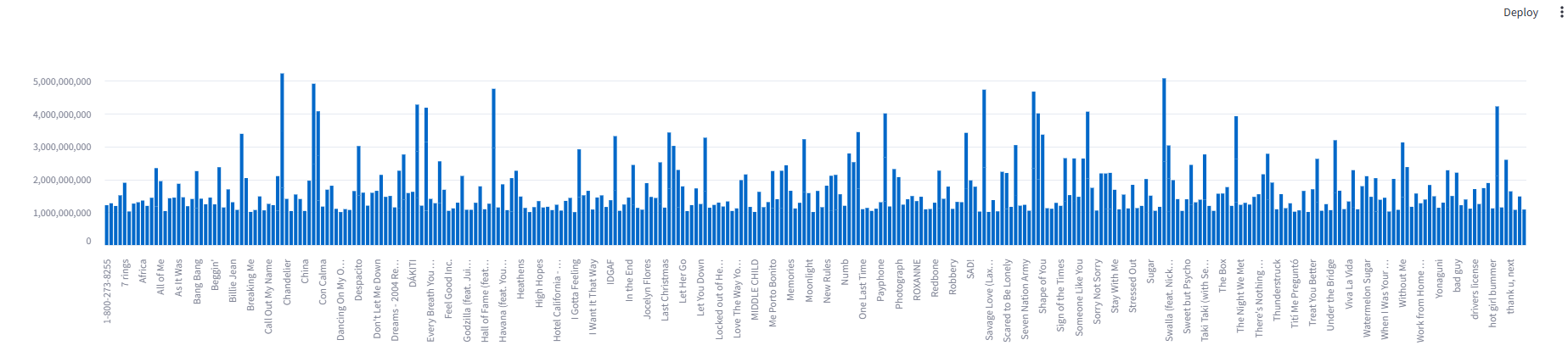
***page\_title*="spotify songs"**

**)**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Track", *inplace*=True)**

**st.bar\_chart(df[df["Stream"] >= 1000000000]["Stream"])**



Vamos usar um select box para filtrar o df

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

***page\_title*="spotify songs"**

**)**

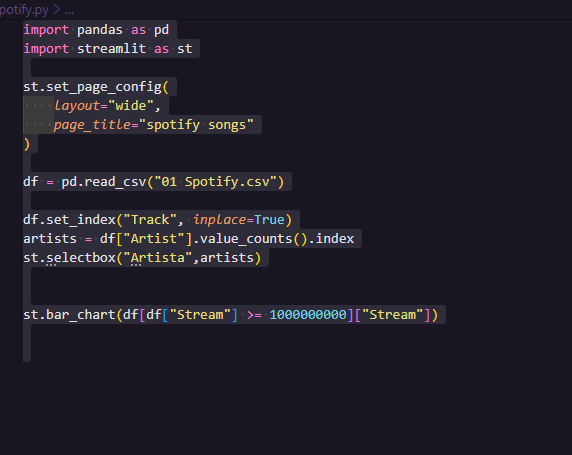
**df = pd.read\_csv("01 Spotify.csv")**

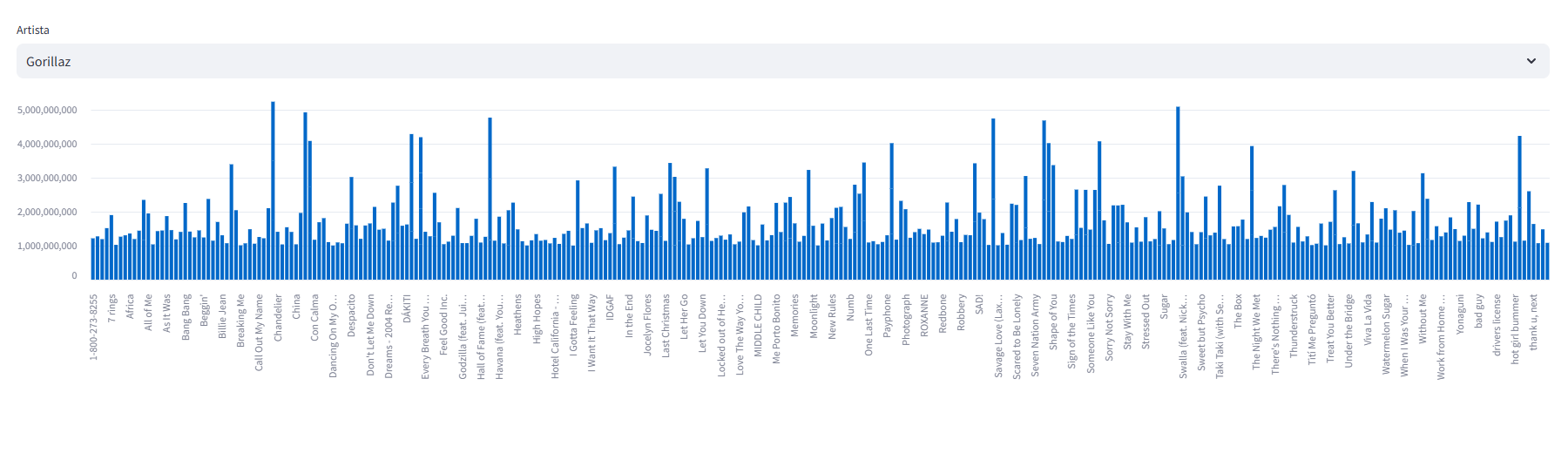
**df.set\_index("Track", *inplace*=True)**

**artists = df["Artist"].value\_counts().index**

**st.selectbox("Artista",artists)**

**st.bar\_chart(df[df["Stream"] >= 1000000000]["Stream"])**

****

****

vamos filtrar só uma banda

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

***page\_title*="spotify songs"**

**)**

**df = pd.read\_csv("01 Spotify.csv")**

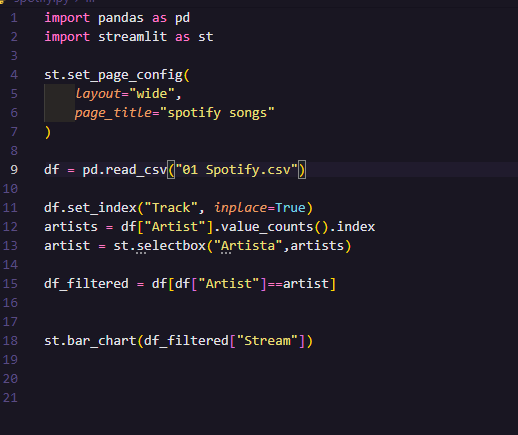
**df.set\_index("Track", *inplace*=True)**

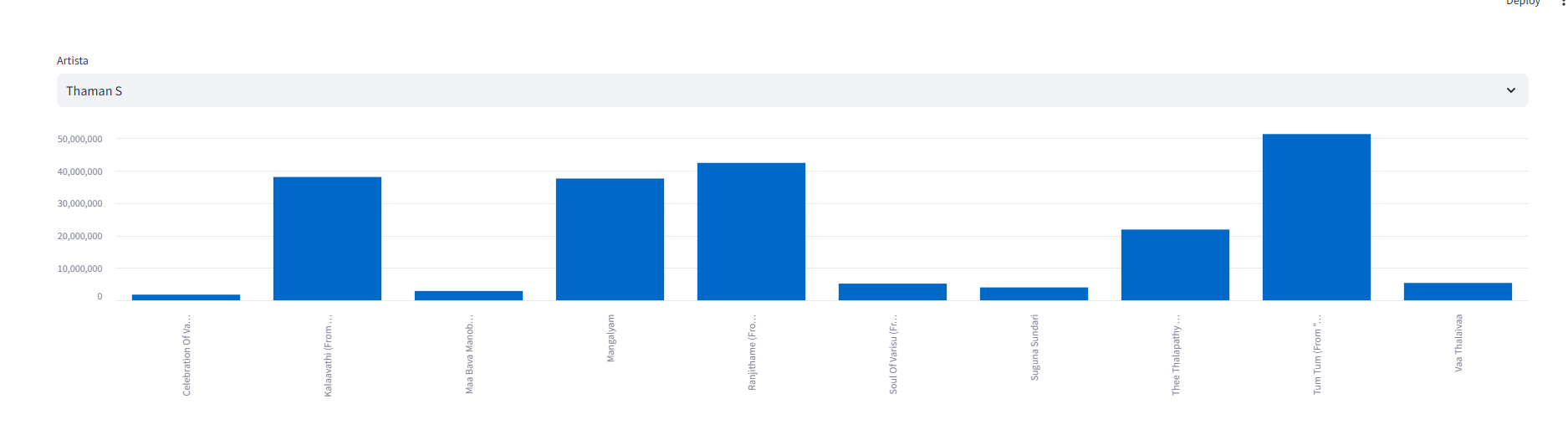
**artists = df["Artist"].value\_counts().index**

**artist = st.selectbox("Artista",artists)**

**df\_filtered = df[df["Artist"]==artist]**

**st.bar\_chart(df\_filtered["Stream"])**

****

****

Poderíamos printar qual o artista selecionado

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

***page\_title*="spotify songs"**

**)**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Track", *inplace*=True)**

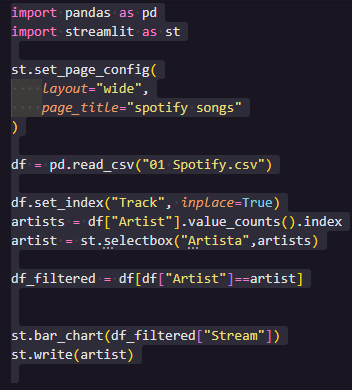
**artists = df["Artist"].value\_counts().index**

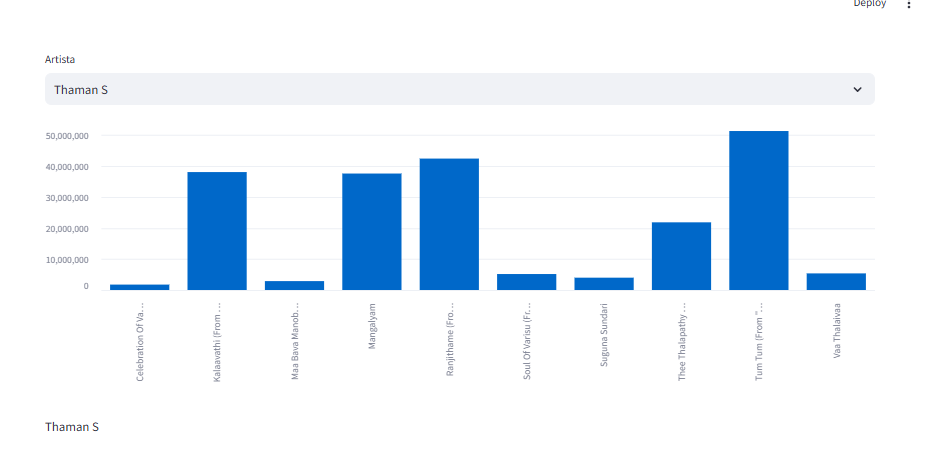
**artist = st.selectbox("Artista",artists)**

**df\_filtered = df[df["Artist"]==artist]**

**st.bar\_chart(df\_filtered["Stream"])**

**st.write(artist)**





Vamos colocar o um checkbox para esconder ou mostrar o Gráfico

import pandas as pd

import streamlit as st

st.set\_page\_config(

*layout*="wide",

*page\_title*="spotify songs"

)

df = pd.read\_csv("01 Spotify.csv")

df.set\_index("Track", *inplace*=True)

artists = df["Artist"].value\_counts().index

artist = st.selectbox("Artista",artists)

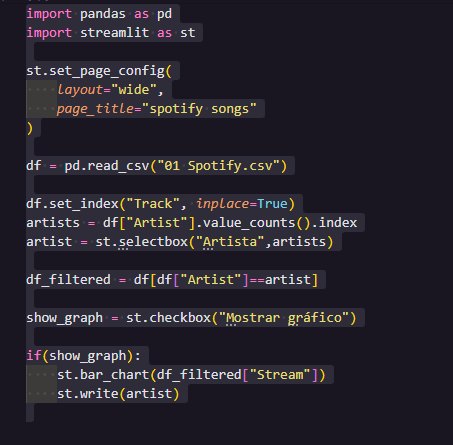
df\_filtered = df[df["Artist"]==artist]

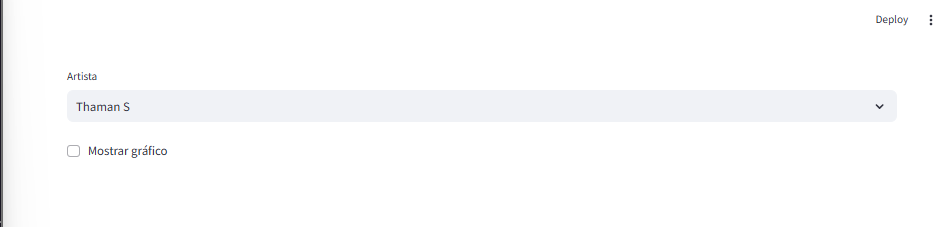
show\_graph = st.checkbox("Mostrar gráfico")

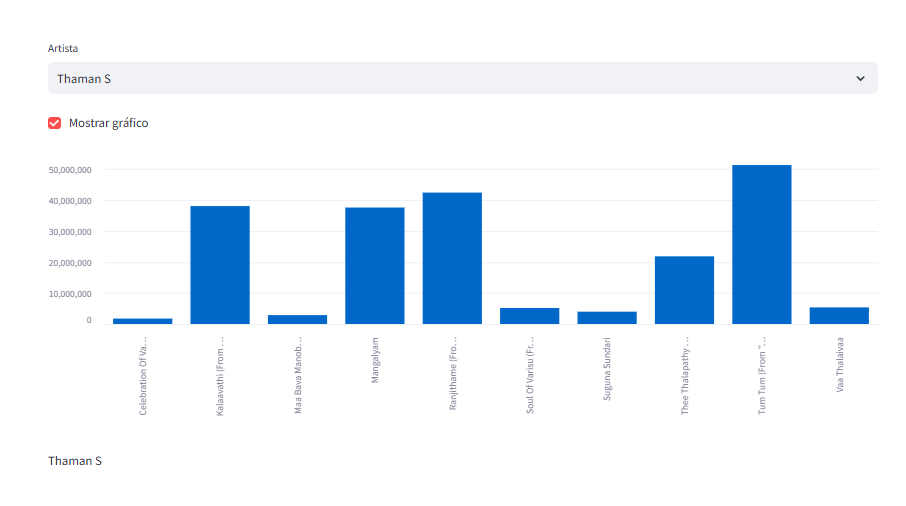
if(show\_graph):

    st.bar\_chart(df\_filtered["Stream"])

    st.write(artist)







Vamos selecionar um artista e um álbum desse artista

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

***page\_title*="spotify songs"**

**)**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Track", *inplace*=True)**

**artists = df["Artist"].value\_counts().index**

**artist = st.selectbox("Artista",artists)**

**df\_filtered = df[df["Artist"]==artist]**

**albuns = df\_filtered["Album"].value\_counts().index**

**album = st.selectbox("Album",albuns)**

**df\_filtered2 = df\_filtered[df\_filtered["Album"]==album]**

**show\_graph = st.checkbox("Mostrar gráfico")**

**if(show\_graph):**

**st.bar\_chart(df\_filtered2["Stream"])**

**st.write(artist)**

****

****

## [Layout](https://hub.asimov.academy/curso/atividade/layout-3/)

Vamos colocar a check box d eartista e álbum na esquerda:

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

***page\_title*="spotify songs"**

**)**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Track", *inplace*=True)**

**artists = df["Artist"].value\_counts().index**

**artist = st.sidebar.selectbox("Artista",artists)**

**df\_filtered = df[df["Artist"]==artist]**

**albuns = df\_filtered["Album"].value\_counts().index**

**album = st.sidebar.selectbox("Album",albuns)**

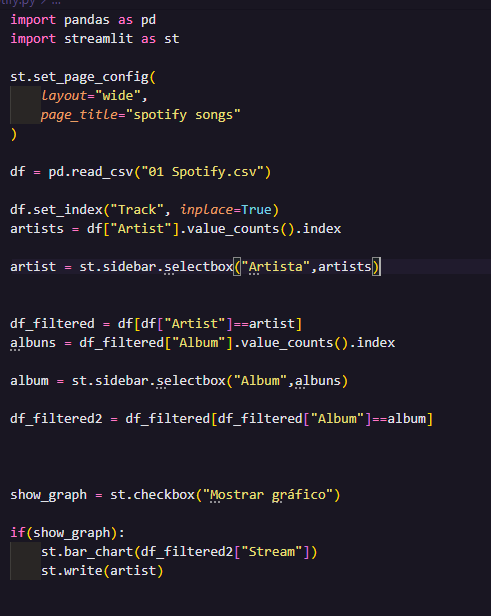
**df\_filtered2 = df\_filtered[df\_filtered["Album"]==album]**

**show\_graph = st.checkbox("Mostrar gráfico")**

**if(show\_graph):**

**st.bar\_chart(df\_filtered2["Stream"])**

**st.write(artist)**



Vamos tirar o chekbox e colcoar duas colunas. Uma para o **Stream** e outr apara o **Danceability**

**import pandas as pd**

**import streamlit as st**

**st.set\_page\_config(**

***layout*="wide",**

***page\_title*="spotify songs"**

**)**

**df = pd.read\_csv("01 Spotify.csv")**

**df.set\_index("Track", *inplace*=True)**

**artists = df["Artist"].value\_counts().index**

**artist = st.sidebar.selectbox("Artista",artists)**

**df\_filtered = df[df["Artist"]==artist]**

**albuns = df\_filtered["Album"].value\_counts().index**

**album = st.sidebar.selectbox("Album",albuns)**

**df\_filtered2 = df\_filtered[df\_filtered["Album"]==album]**

**# show\_graph = st.checkbox("Mostrar gráfico")**

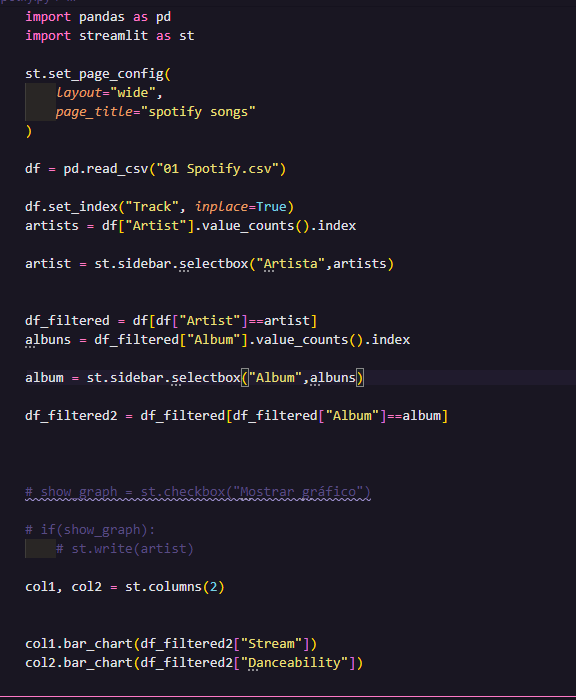
**# if(show\_graph):**

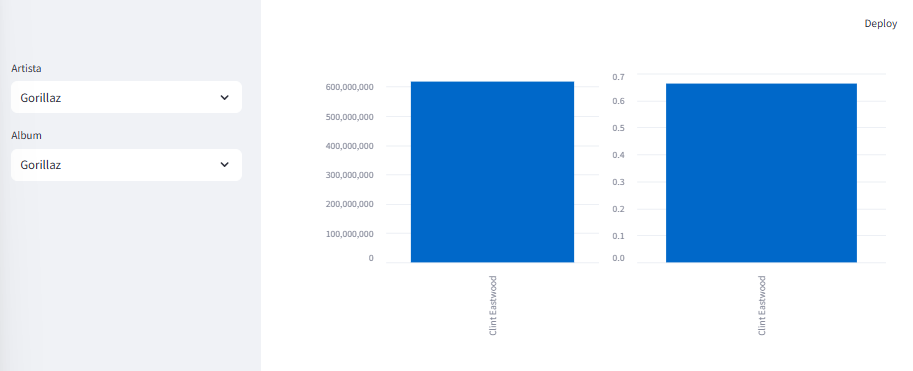
**# st.write(artist)**

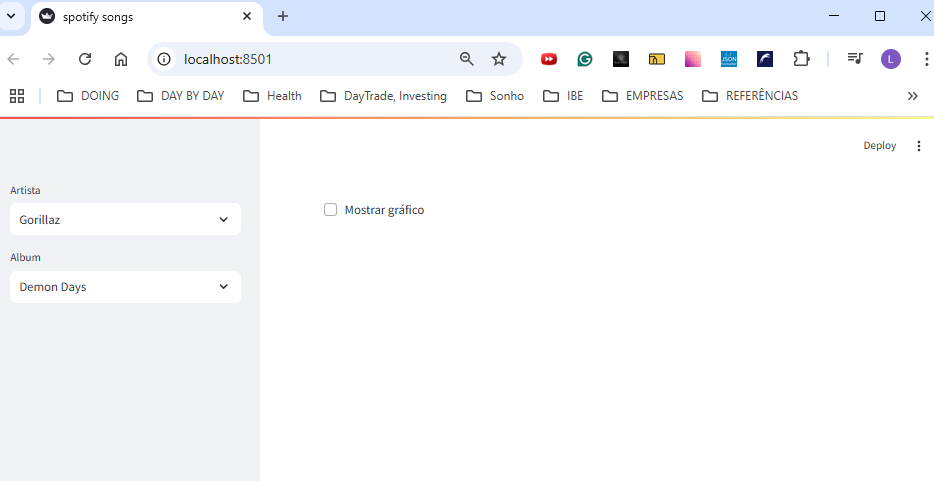
**col1, col2 = st.columns(2)**

**col1.bar\_chart(df\_filtered2["Stream"])**

**col2.bar\_chart(df\_filtered2["Danceability"])**

****

****



Se quisermos divifir pela quantidade de despeaço ocupado, podemos passa rum array com as percentagens de espeço para cada coluna

import pandas as pd

import streamlit as st

st.set\_page\_config(

*layout*="wide",

*page\_title*="spotify songs"

)

df = pd.read\_csv("01 Spotify.csv")

df.set\_index("Track", *inplace*=True)

artists = df["Artist"].value\_counts().index

artist = st.sidebar.selectbox("Artista",artists)

df\_filtered = df[df["Artist"]==artist]

albuns = df\_filtered["Album"].value\_counts().index

album = st.sidebar.selectbox("Album",albuns)

df\_filtered2 = df\_filtered[df\_filtered["Album"]==album]

# show\_graph = st.checkbox("Mostrar gráfico")

# if(show\_graph):

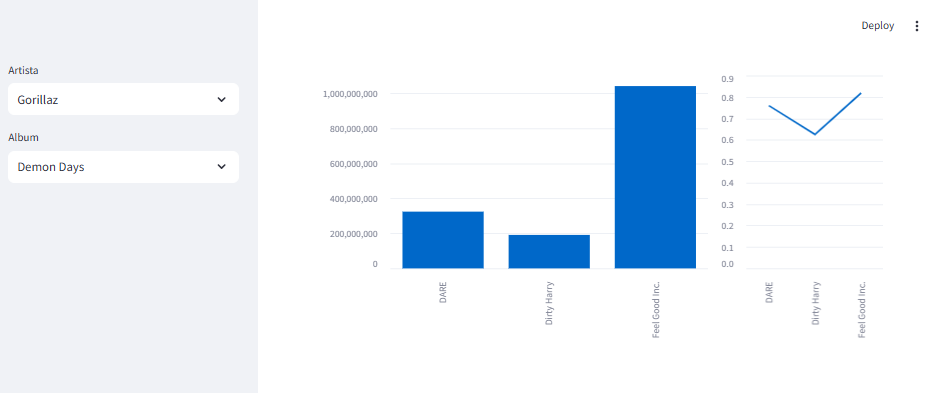
    # st.write(artist)

**col1, col2 = st.columns([0.7,0.3])**

col1.bar\_chart(df\_filtered2["Stream"])

col2.line\_chart(df\_filtered2["Danceability"])

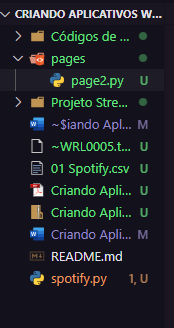


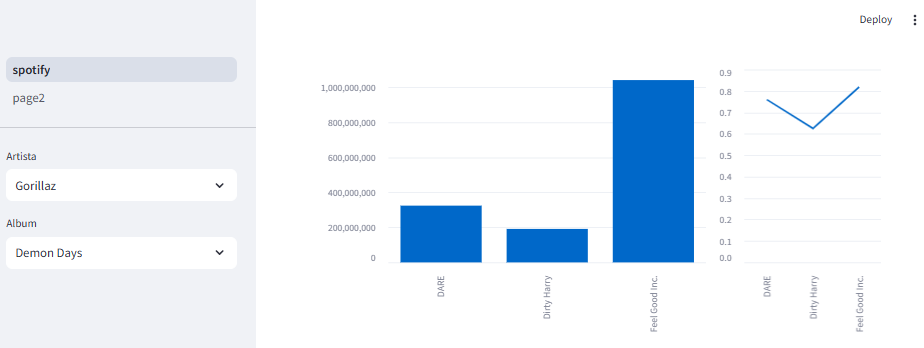


## [Cacheamento e Multipages](https://hub.asimov.academy/curso/atividade/cacheamento-e-multipages/)

Caching multiupages

Para termos muitas página temo eu criar uma pasta chamada **pages** e um arquivo nela:







Para que os dados de uma página sejam compartilhados para outra página, usamos **session\_state**

**SPOTIFY.PY**

import pandas as pd

import streamlit as st

st.set\_page\_config(

*layout*="wide",

*page\_title*="spotify songs"

)

df = pd.read\_csv("01 Spotify.csv")

**st.session\_state['df\_spotify'] = df**

df.set\_index("Track", *inplace*=True)

artists = df["Artist"].value\_counts().index

artist = st.sidebar.selectbox("Artista",artists)

df\_filtered = df[df["Artist"]==artist]

albuns = df\_filtered["Album"].value\_counts().index

album = st.sidebar.selectbox("Album",albuns)

df\_filtered2 = df\_filtered[df\_filtered["Album"]==album]

# show\_graph = st.checkbox("Mostrar gráfico")

# if(show\_graph):

    # st.write(artist)

col1, col2 = st.columns([0.7,0.3])

col1.bar\_chart(df\_filtered2["Stream"])

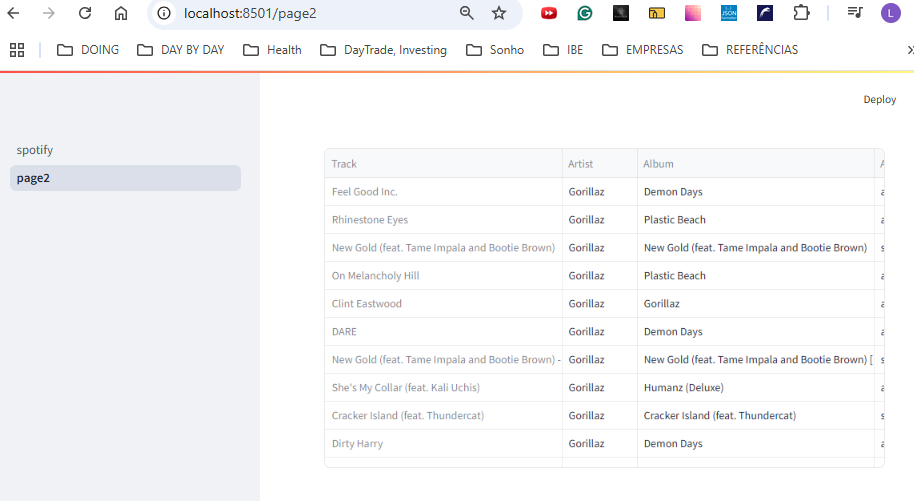
col2.line\_chart(df\_filtered2["Danceability"])

**PAGE2.PY**

import pandas as pd

import streamlit as st

st.session\_state['df\_spotify']



Podemos passar uma operação muito pessada para dentrodo cache numa função:

import pandas as pd

import streamlit as st

st.set\_page\_config(

*layout*="wide",

*page\_title*="spotify songs"

)

**@st.cache\_data**

**def load\_data():**

**df = pd.read\_csv("01 Spotify.csv")**

**return df**

**st.session\_state['df\_spotify'] = load\_data()**

**df = load\_data()**

df.set\_index("Track", *inplace*=True)

artists = df["Artist"].value\_counts().index

artist = st.sidebar.selectbox("Artista",artists)

df\_filtered = df[df["Artist"]==artist]

albuns = df\_filtered["Album"].value\_counts().index

album = st.sidebar.selectbox("Album",albuns)

df\_filtered2 = df\_filtered[df\_filtered["Album"]==album]

# show\_graph = st.checkbox("Mostrar gráfico")

# if(show\_graph):

    # st.write(artist)

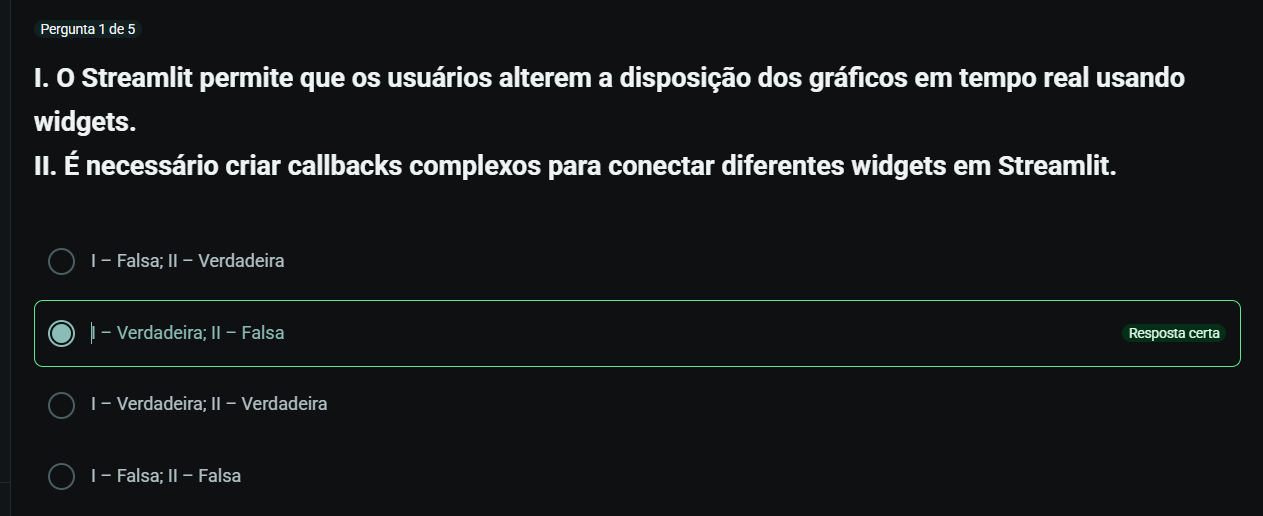
col1, col2 = st.columns([0.7,0.3])

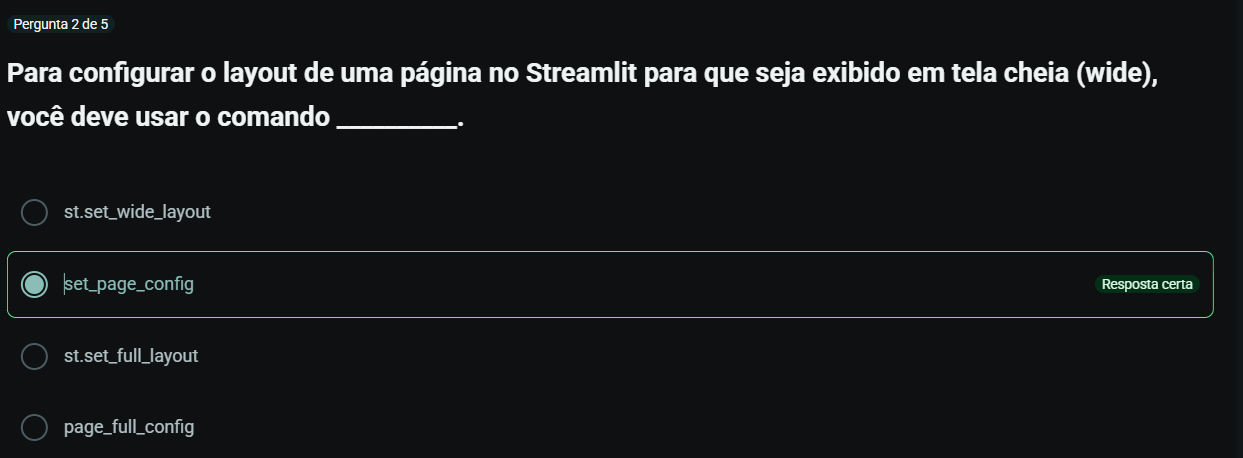
col1.bar\_chart(df\_filtered2["Stream"])

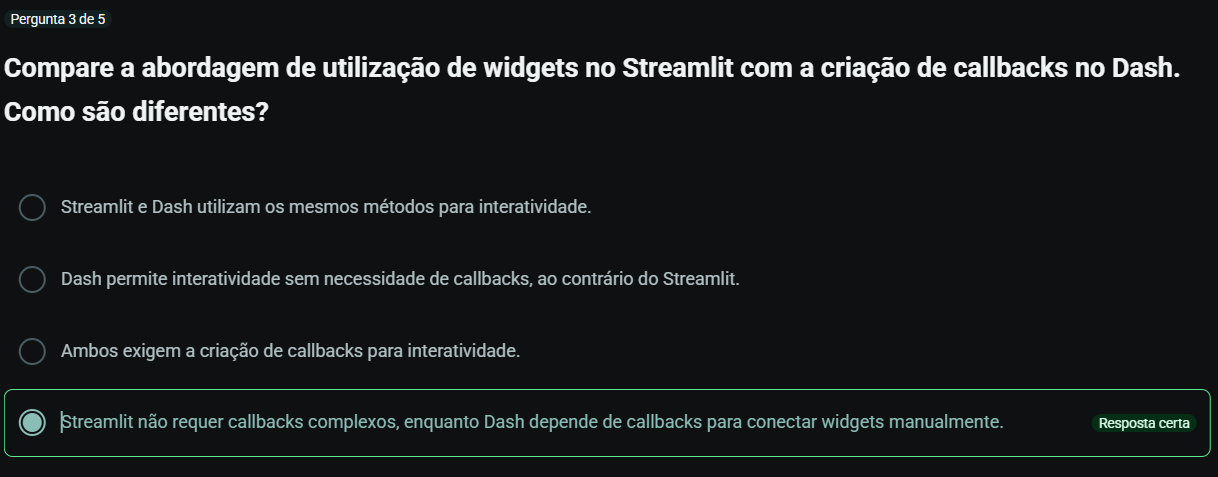
col2.line\_chart(df\_filtered2["Danceability"])

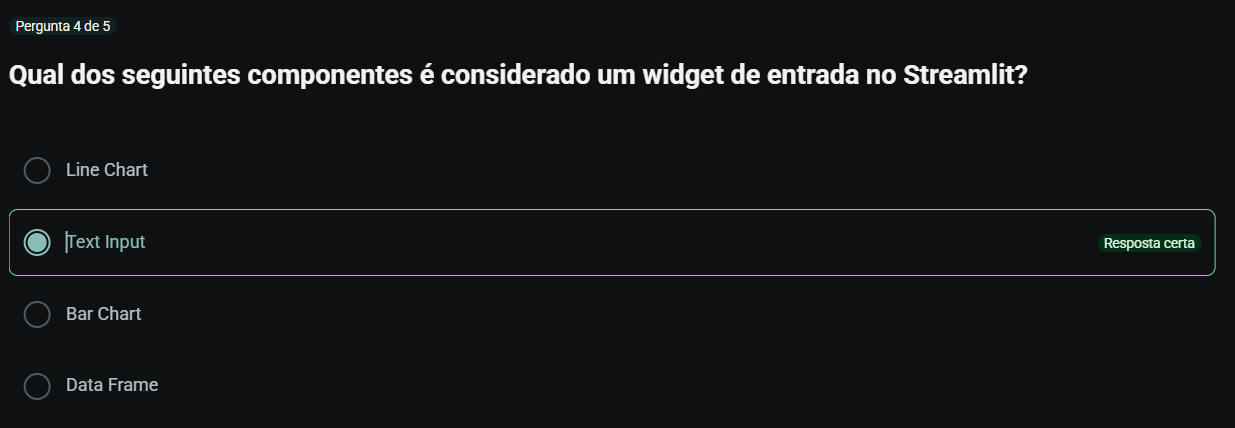
## [Componentes adicionais](https://hub.asimov.academy/curso/atividade/componentes-adicionais/)

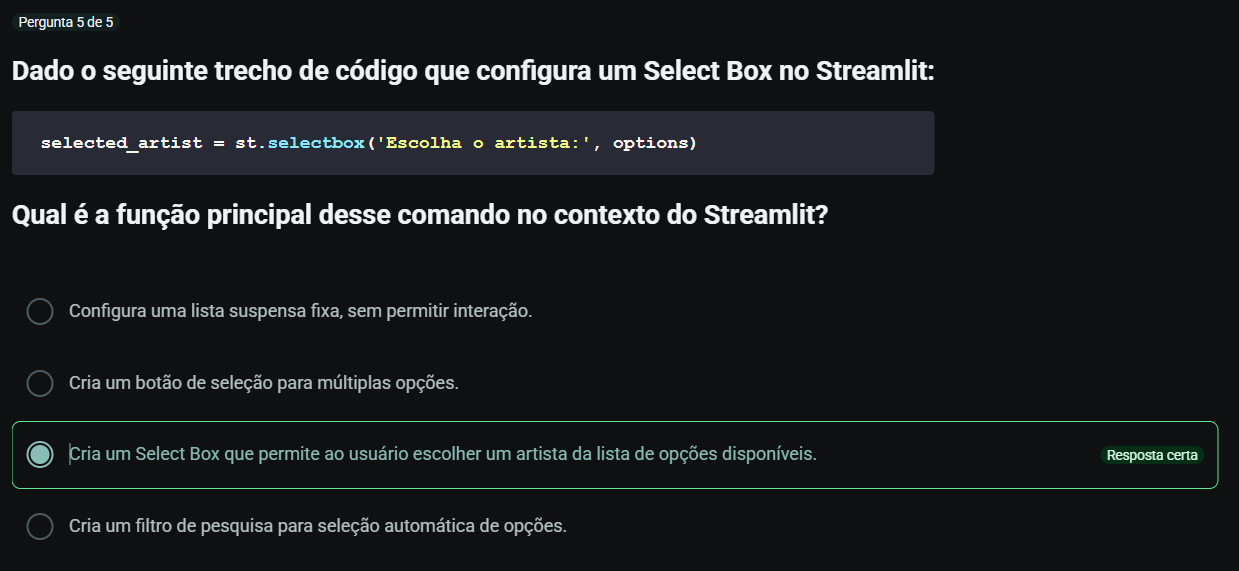
## [Quiz – Como customizar seu aplicativo](https://hub.asimov.academy/curso/atividade/quiz-como-customizar-seu-aplicativo/)









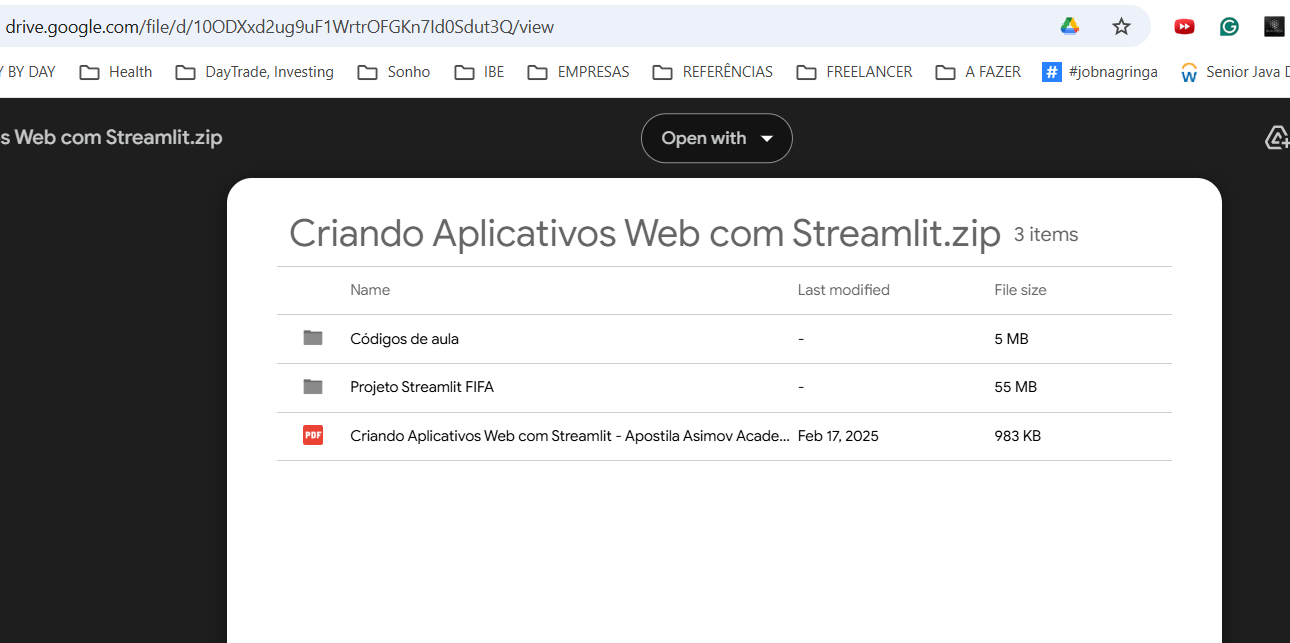


# Análise de dados da FIFA com Streamlit

## [Dash FIFA 2023](https://hub.asimov.academy/curso/atividade/dash-fifa-2023/)

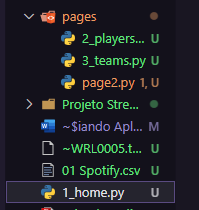
Pegar o dataset

<https://drive.google.com/file/d/10ODXxd2ug9uF1WrtrOFGKn7Id0Sdut3Q/view>



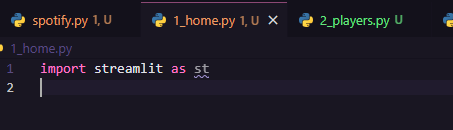
Para definir as ordens das páginas usamos o seguinte padrão

**1\_nome\_da\_pagina.py**

****

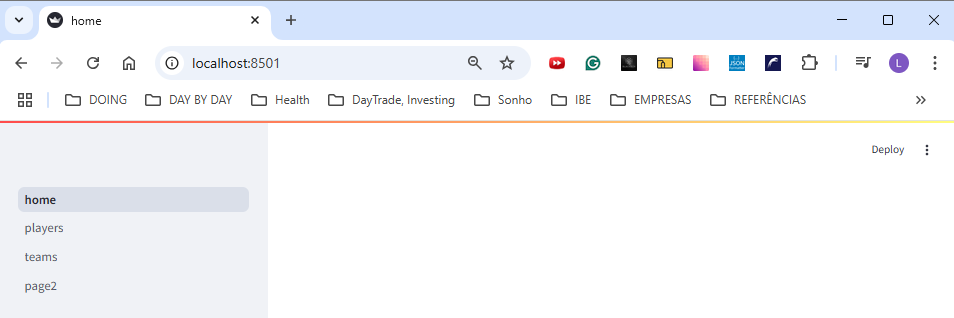
Para todos eles, vamos importar o **streamlit**

**import streamlit as st**

****

para rodar vamos executar

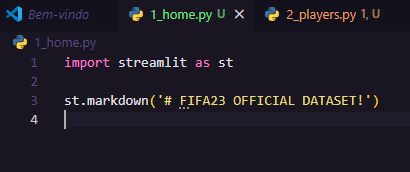
**streamlit run 1\_home.py**

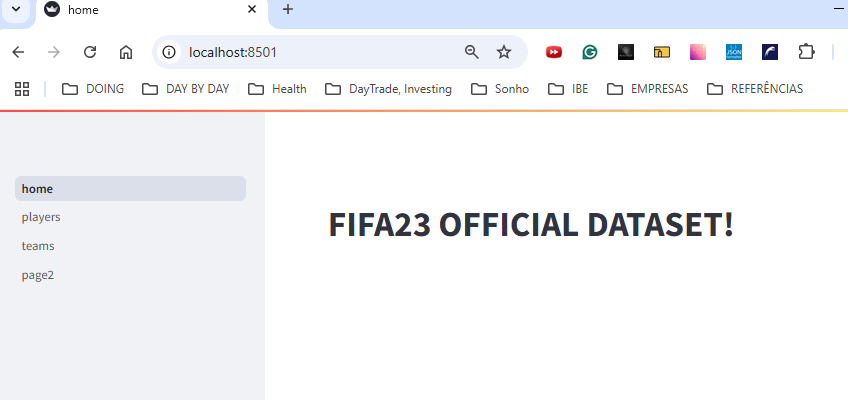
****

Vamos colocar um markdown na home

import streamlit as st

**st.markdown('# FIFA23 OFFICIAL DATASET! ')**





Adicione

**import webbrowser**

**btn = st.button("Acesse os dados no Kaggle")**

**if btn:**

**webbrowser.open\_new\_tab("https://www.kaggle.com/datasets/kevwesophia/fifa23-official-datasetclean-data")**

import webbrowser

import streamlit as st

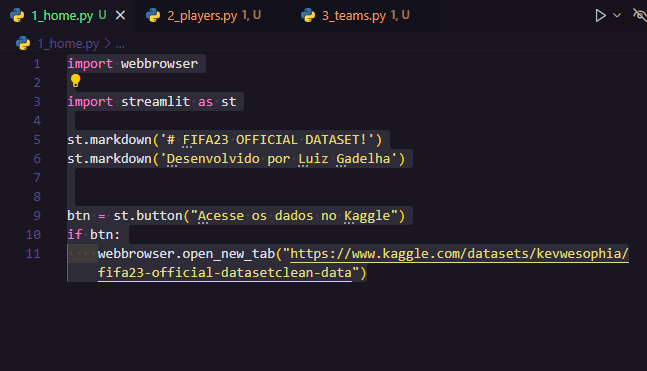
st.markdown('# FIFA23 OFFICIAL DATASET!')

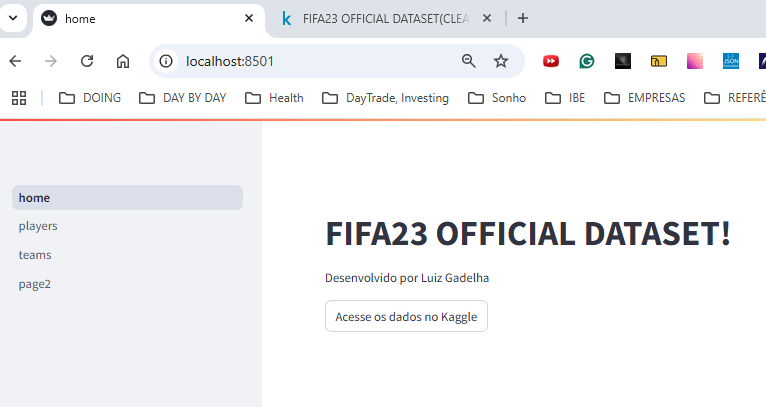
st.markdown('Desenvolvido por Luiz Gadelha')

btn = st.button("Acesse os dados no Kaggle")

if btn:

    webbrowser.open\_new\_tab("https://www.kaggle.com/datasets/kevwesophia/fifa23-official-datasetclean-data")





Vamos adicionar

st.markdown(

    """

    O conjunto de dados

    de jogadores de futebol de 2017 a 2023 fornece informações

    abrangentes sobre jogadores de futebol profissionais.

    O conjunto de dados contém uma ampla gama de atributos, incluindo dados demográficos

    do jogador, características físicas, estatísticas de jogo, detalhes do contrato e

    afiliações de clubes.

    Com \*\*mais de 17.000 registros\*\*, este conjunto de dados oferece um recurso valioso para

    analistas de futebol, pesquisadores e entusiastas interessados em explorar vários

    aspectos do mundo do futebol, pois permite estudar atributos de jogadores, métricas de

    desempenho, avaliação de mercado, análise de clubes, posicionamento de jogadores e

    desenvolvimento do jogador ao longo do tempo.

"""

)

import webbrowser

import streamlit as st

st.markdown('# FIFA23 OFFICIAL DATASET!')

st.markdown('Desenvolvido por Luiz Gadelha')

btn = st.button("Acesse os dados no Kaggle")

if btn:

    webbrowser.open\_new\_tab("https://www.kaggle.com/datasets/kevwesophia/fifa23-official-datasetclean-data")

**st.markdown(**

**"""**

**O conjunto de dados**

**de jogadores de futebol de 2017 a 2023 fornece informações**

**abrangentes sobre jogadores de futebol profissionais.**

**O conjunto de dados contém uma ampla gama de atributos, incluindo dados demográficos**

**do jogador, características físicas, estatísticas de jogo, detalhes do contrato e**

**afiliações de clubes.**

**Com \*\*mais de 17.000 registros\*\*, este conjunto de dados oferece um recurso valioso para**

**analistas de futebol, pesquisadores e entusiastas interessados em explorar vários**

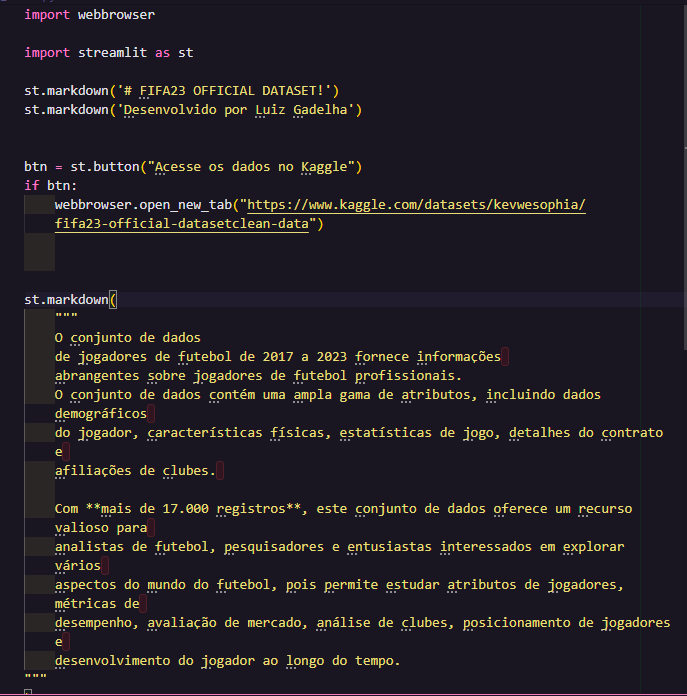
**aspectos do mundo do futebol, pois permite estudar atributos de jogadores, métricas de**

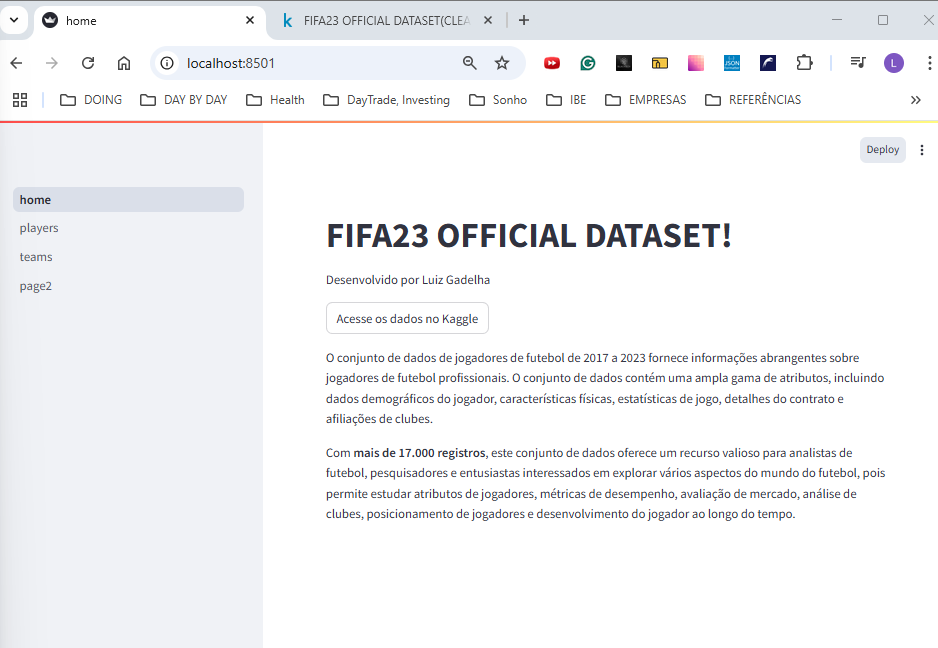
**desempenho, avaliação de mercado, análise de clubes, posicionamento de jogadores e**

**desenvolvimento do jogador ao longo do tempo.**

**"""**

**)**

****



Vamos carregar os dados se não estão na sessão

import pandas as pd

from datetime import datetime

if "data" not in st.session\_state:

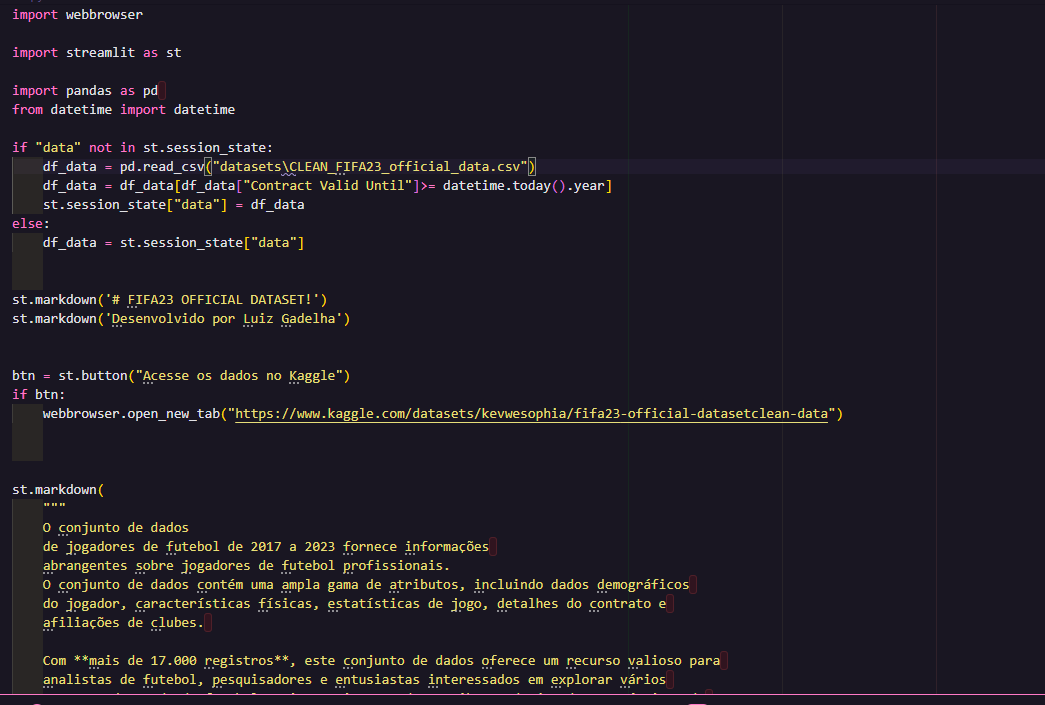
    df\_data = pd.read\_csv("datasets\CLEAN\_FIFA23\_official\_data.csv")

    df\_data = df\_data[df\_data["Contract Valid Until"]>= datetime.today().year]

    st.session\_state["data"] = df\_data

else:

    df\_data = st.session\_state["data"]



Vamos selecionar apenas jogadores com salario valido e ordernar

import webbrowser

import streamlit as st

**import pandas as pd**

**from datetime import datetime**

**if "data" not in st.session\_state:**

**df\_data = pd.read\_csv("datasets\CLEAN\_FIFA23\_official\_data.csv")**

**df\_data = df\_data[df\_data["Contract Valid Until"]>= datetime.today().year]**

**df\_data = df\_data[df\_data["Value(£)"] > 0]**

**df\_data = df\_data.sort\_values(*by*="Overall", *ascending*=False)**

**st.session\_state["data"] = df\_data**

**else:**

**df\_data = st.session\_state["data"]**

st.markdown('# FIFA23 OFFICIAL DATASET!')

st.markdown('Desenvolvido por Luiz Gadelha')

btn = st.button("Acesse os dados no Kaggle")

if btn:

    webbrowser.open\_new\_tab("https://www.kaggle.com/datasets/kevwesophia/fifa23-official-datasetclean-data")

st.markdown(

    """

    O conjunto de dados

    de jogadores de futebol de 2017 a 2023 fornece informações

    abrangentes sobre jogadores de futebol profissionais.

    O conjunto de dados contém uma ampla gama de atributos, incluindo dados demográficos

    do jogador, características físicas, estatísticas de jogo, detalhes do contrato e

    afiliações de clubes.

    Com \*\*mais de 17.000 registros\*\*, este conjunto de dados oferece um recurso valioso para

    analistas de futebol, pesquisadores e entusiastas interessados em explorar vários

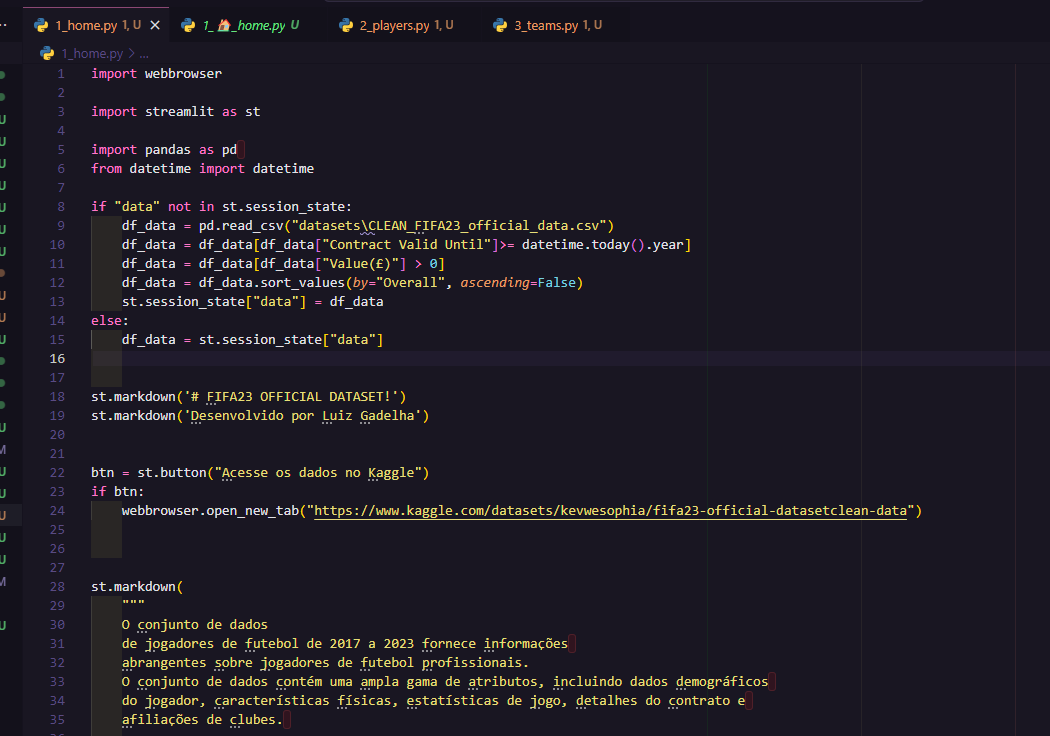
    aspectos do mundo do futebol, pois permite estudar atributos de jogadores, métricas de

    desempenho, avaliação de mercado, análise de clubes, posicionamento de jogadores e

    desenvolvimento do jogador ao longo do tempo.

"""

)

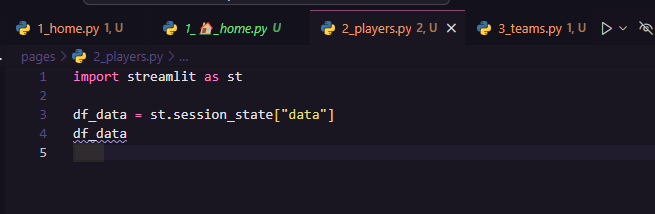


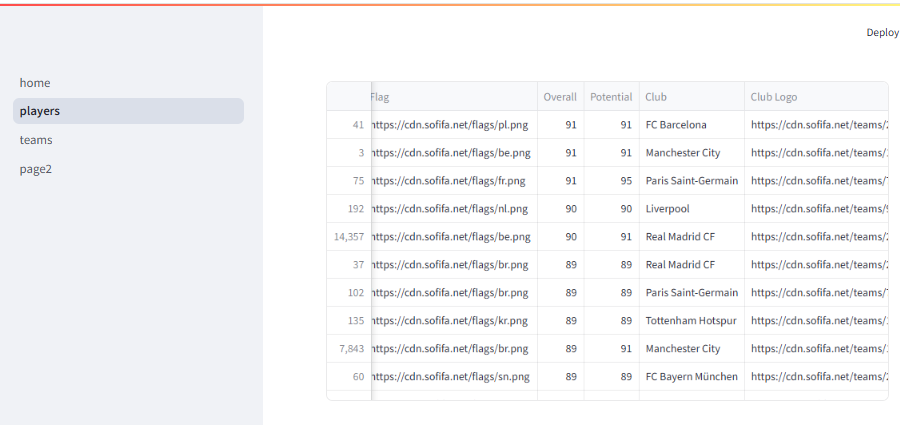
Lendo no players:

**import streamlit as st**

**df\_data = st.session\_state["data"]**

**df\_data**

****

****

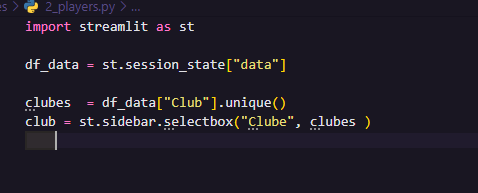
Vamos colocar um seletor dos clubes e um seletor dos jogadores nesse clube

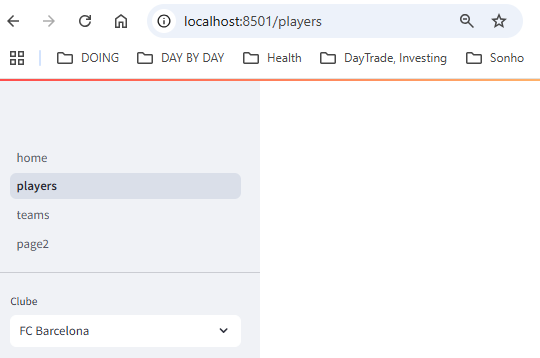
**import streamlit as st**

**df\_data = st.session\_state["data"]**

**clubes  = df\_data["Club"].unique()**

**club = st.sidebar.selectbox("Clube", clubes )**

****

****

Vamos usar esse clube selecionado para pegar os jogadores

import streamlit as st

df\_data = st.session\_state["data"]

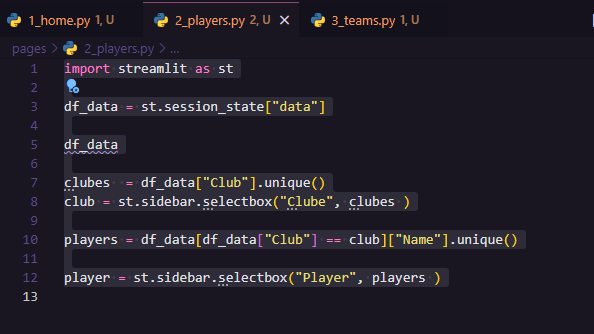
df\_data

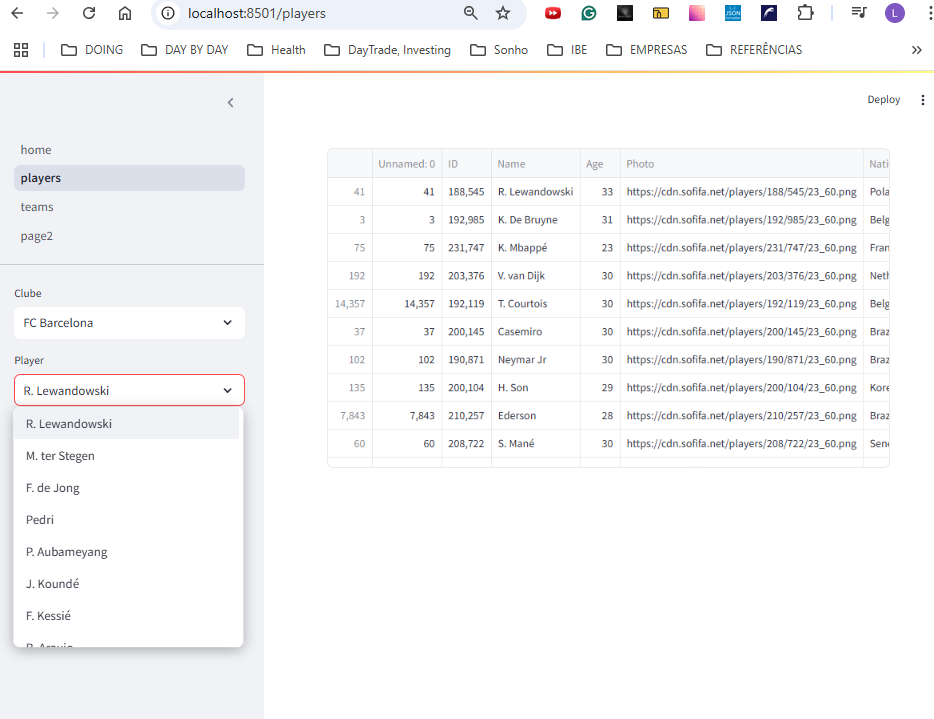
clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

players = df\_data[df\_data["Club"] == club]["Name"].unique()

player = st.sidebar.selectbox("Player", players )





Vamos pegar a linha com o nome do jogador

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

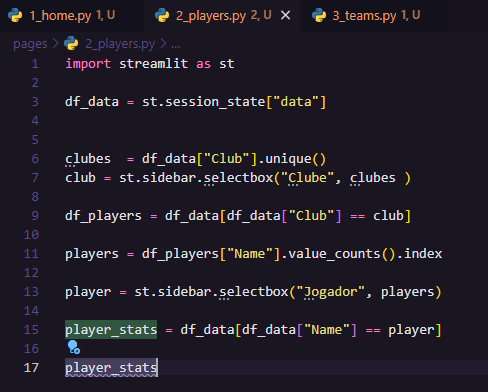
df\_players = df\_data[df\_data["Club"] == club]

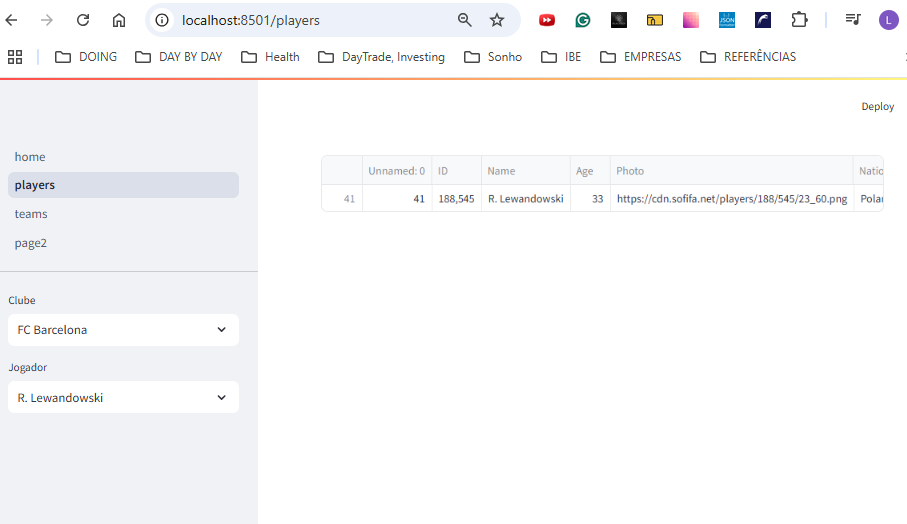
players = df\_players["Name"].value\_counts().index

player = st.sidebar.selectbox("Jogador", players)

player\_stats = df\_data[df\_data["Name"] == player]

player\_stats





Usando a primeira aparição do registro

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

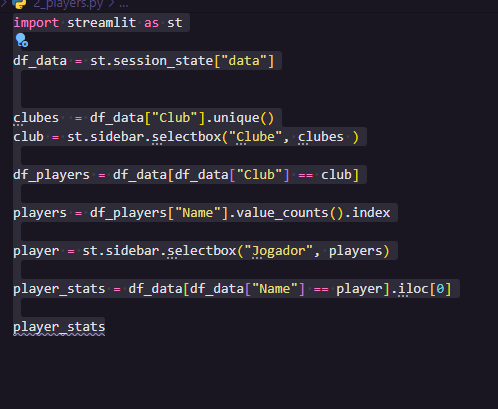
df\_players = df\_data[df\_data["Club"] == club]

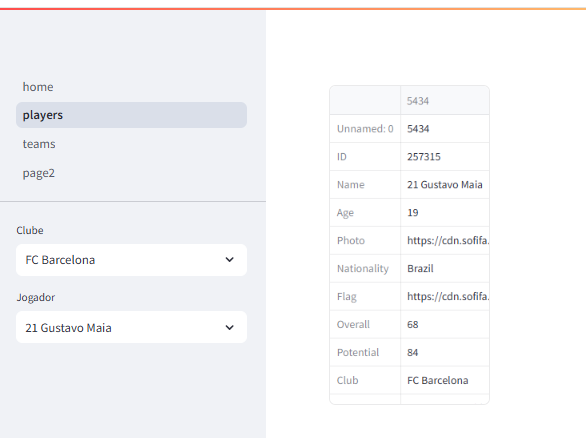
players = df\_players["Name"].value\_counts().index

player = st.sidebar.selectbox("Jogador", players)

player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

player\_stats





Primeiramente vamos colocar a foto usando a Coluna **Photo**

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

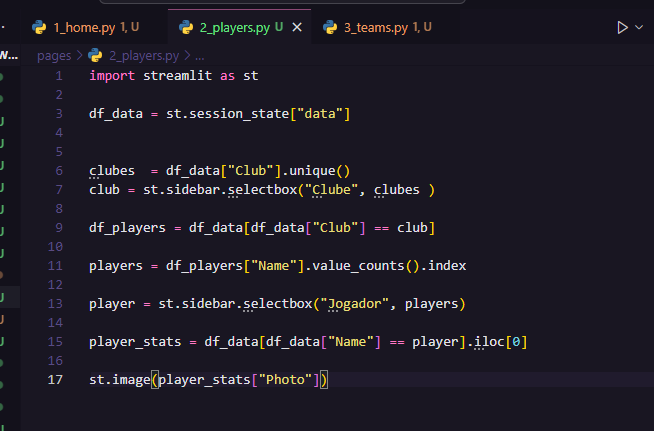
df\_players = df\_data[df\_data["Club"] == club]

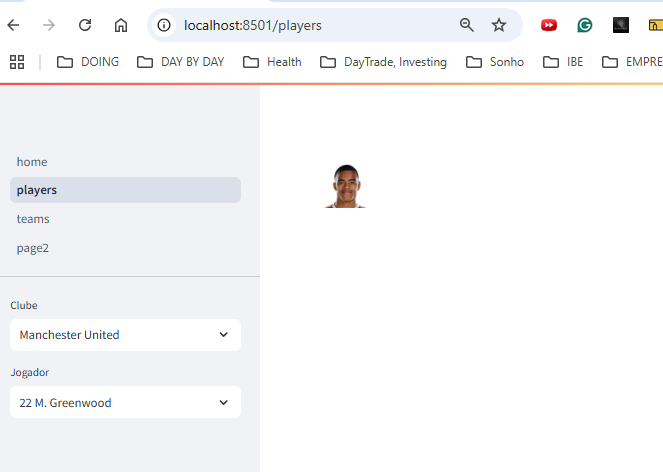
players = df\_players["Name"].value\_counts().index

player = st.sidebar.selectbox("Jogador", players)

player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

**st.image(player\_stats["Photo"])**





Vamos colcoar o título com o nome do jogador

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_players = df\_data[df\_data["Club"] == club]

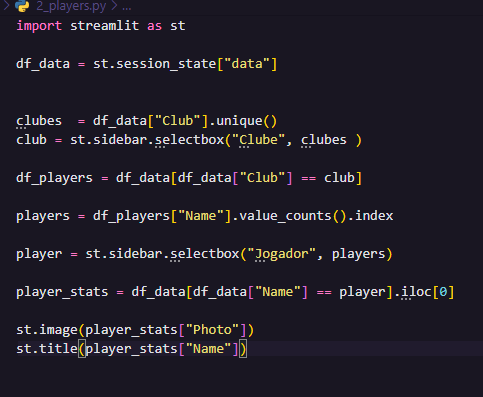
players = df\_players["Name"].value\_counts().index

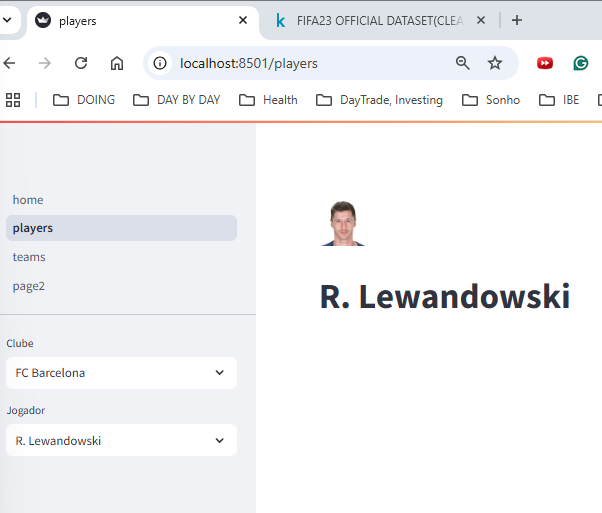
player = st.sidebar.selectbox("Jogador", players)

player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

st.image(player\_stats["Photo"])

**st.title(player\_stats["Name"])**

****

****

vamos colocar o nome do clube do jogador

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_players = df\_data[df\_data["Club"] == club]

players = df\_players["Name"].value\_counts().index

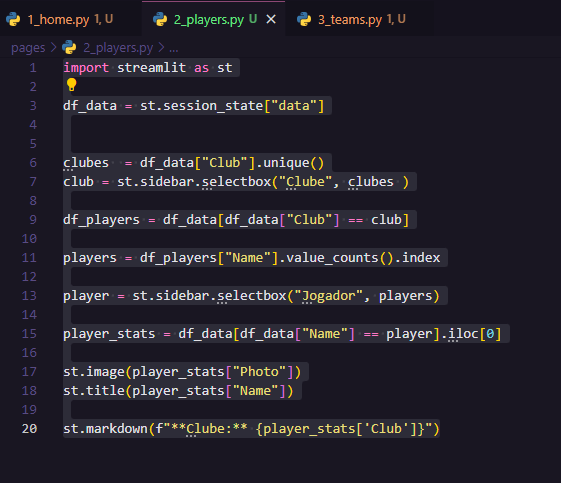
player = st.sidebar.selectbox("Jogador", players)

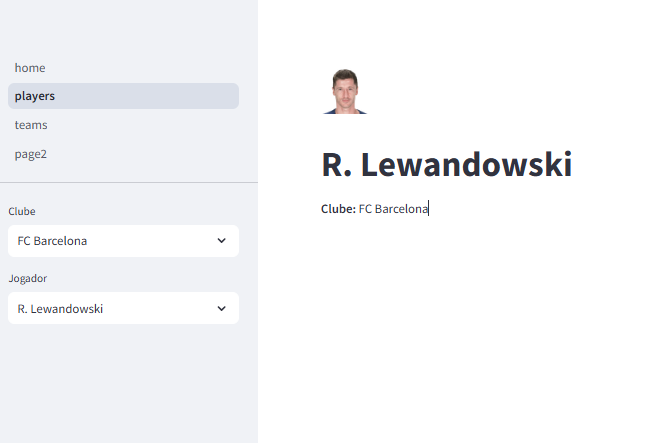
player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

st.image(player\_stats["Photo"])

st.title(player\_stats["Name"])

**st.markdown(f"\*\*Clube:\*\* {player\_stats['Club']}")**

****

****

vamos colocar a posição

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_players = df\_data[df\_data["Club"] == club]

players = df\_players["Name"].value\_counts().index

player = st.sidebar.selectbox("Jogador", players)

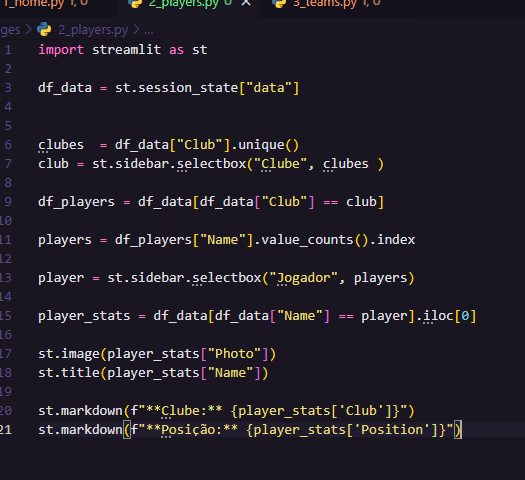
player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

st.image(player\_stats["Photo"])

st.title(player\_stats["Name"])

st.markdown(f"\*\*Clube:\*\* {player\_stats['Club']}")

**st.markdown(f"\*\*Posição:\*\* {player\_stats['Position']}")**

****

****

Vamos colocar 4 colunas para Age, Heigth em metros, Weigth em kg e Overral

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_players = df\_data[df\_data["Club"] == club]

players = df\_players["Name"].value\_counts().index

player = st.sidebar.selectbox("Jogador", players)

player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

st.image(player\_stats["Photo"])

st.title(player\_stats["Name"])

st.markdown(f"\*\*Clube:\*\* {player\_stats['Club']}")

st.markdown(f"\*\*Posição:\*\* {player\_stats['Position']}")

**col1, col2, col3, col4 = st.columns(4)**

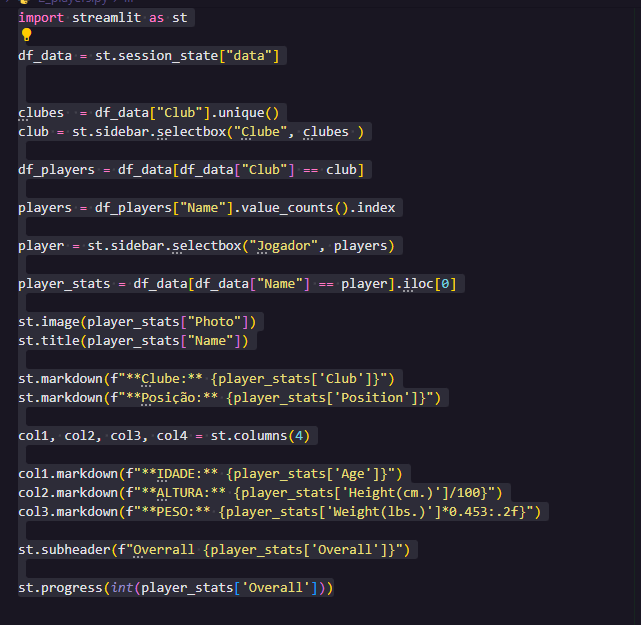
**col1.markdown(f"\*\*IDADE:\*\* {player\_stats['Age']}")**

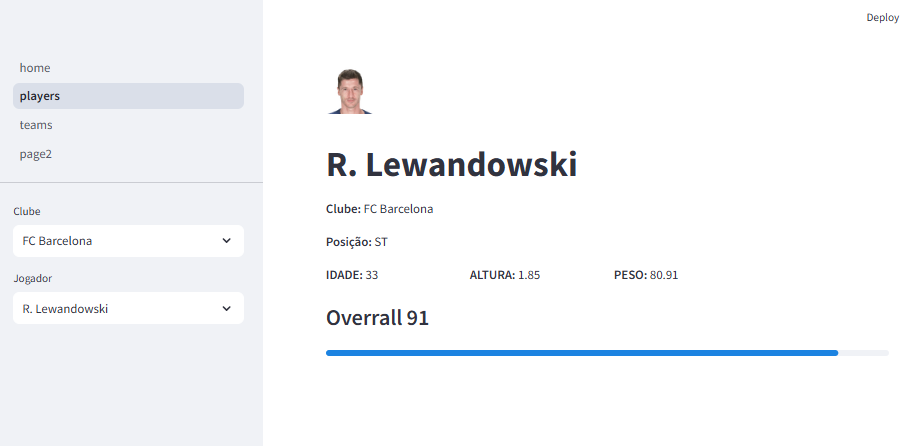
**col2.markdown(f"\*\*ALTURA:\*\* {player\_stats['Height(cm.)']/100}")**

**col3.markdown(f"\*\*PESO:\*\* {player\_stats['Weight(lbs.)']\*0.453:.2f}")**

**st.subheader(f"Overrall {player\_stats['Overall']}")**

**st.progress(*int*(player\_stats['Overall']))**

****

****

Vamos colocar o valor de mercador, remuneração semanal e Cláusula de rescisão

import streamlit as st

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_players = df\_data[df\_data["Club"] == club]

players = df\_players["Name"].value\_counts().index

player = st.sidebar.selectbox("Jogador", players)

player\_stats = df\_data[df\_data["Name"] == player].iloc[0]

st.image(player\_stats["Photo"])

st.title(player\_stats["Name"])

st.markdown(f"\*\*Clube:\*\* {player\_stats['Club']}")

st.markdown(f"\*\*Posição:\*\* {player\_stats['Position']}")

col1, col2, col3, col4 = st.columns(4)

col1.markdown(f"\*\*IDADE:\*\* {player\_stats['Age']}")

col2.markdown(f"\*\*ALTURA:\*\* {player\_stats['Height(cm.)']/100}")

col3.markdown(f"\*\*PESO:\*\* {player\_stats['Weight(lbs.)']\*0.453:.2f}")

st.subheader(f"Overrall {player\_stats['Overall']}")

st.progress(*int*(player\_stats['Overall']))

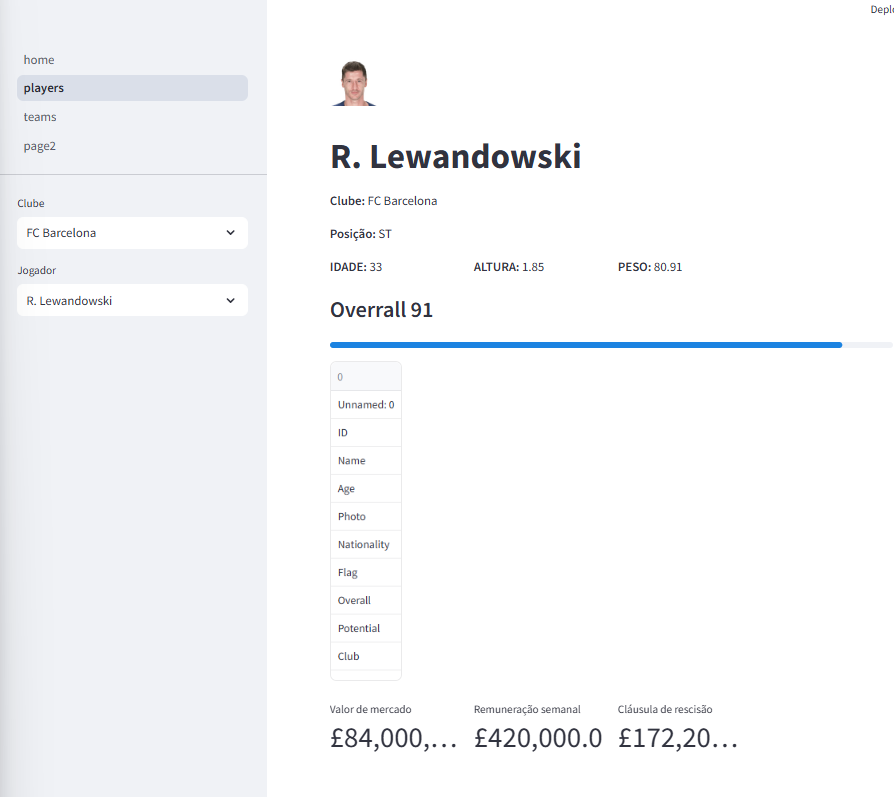
df\_data.columns

**col1, col2, col3, col4 = st.columns(4)**

**col1.metric(*label*="Valor de mercado", *value*=f"£{player\_stats['Value(£)']:,}")**

**col2.metric(*label*="Remuneração semanal", *value*=f"£{player\_stats['Wage(£)']:,}")**

**col3.metric(*label*="Cláusula de rescisão", *value*=f"£{player\_stats['Release Clause(£)']:,}")**

****

Vamos colocar a configuração de wide

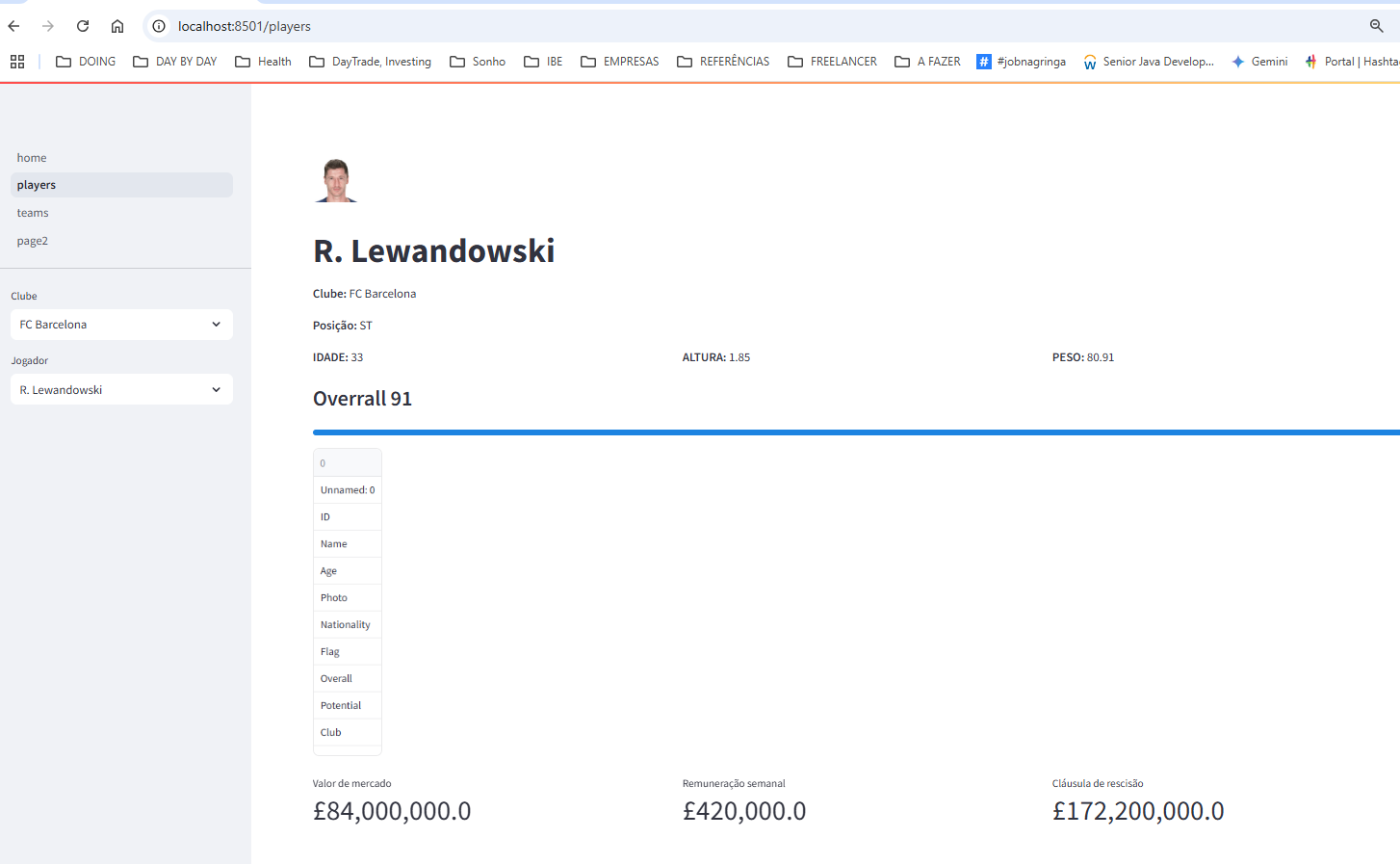
**st.set\_page\_config(**

***page\_title*="Players",**

***page\_icon*="🏃🏼",**

***layout*="wide"**

**)**



Vamos configurar a apágina de timaes

Primeiramente vamos pegar os dados do astate

**df\_data = st.session\_state["data"]**

vamos pegar os clubes

**clubes  = df\_data["Club"].unique()**

**club = st.sidebar.selectbox("Clube", clubes )**

Vamos colocar o símboloe o nome

import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

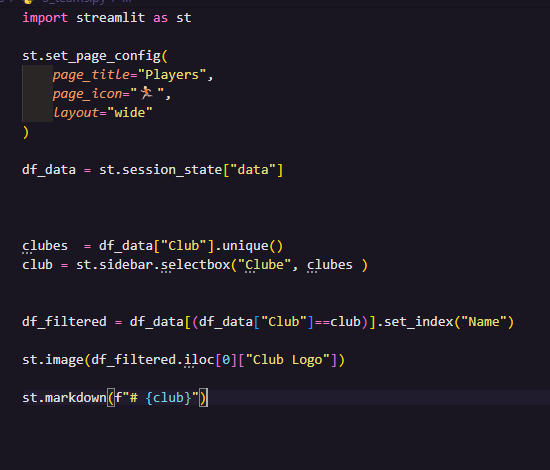
clubes  = df\_data["Club"].unique()

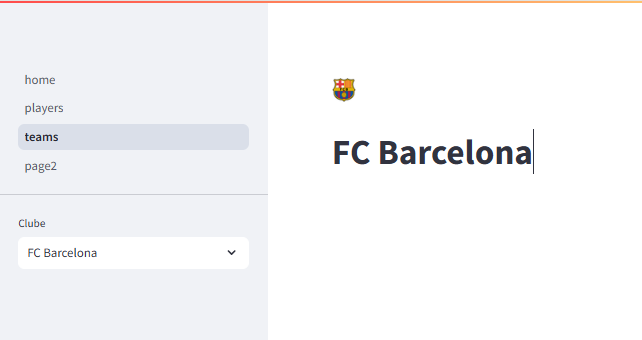
club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

**st.image(df\_filtered.iloc[0]["Club Logo"])**

**st.markdown(f"# {club}")**





Vamos colocar no array todos colunas que desejamos

**columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']**

import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

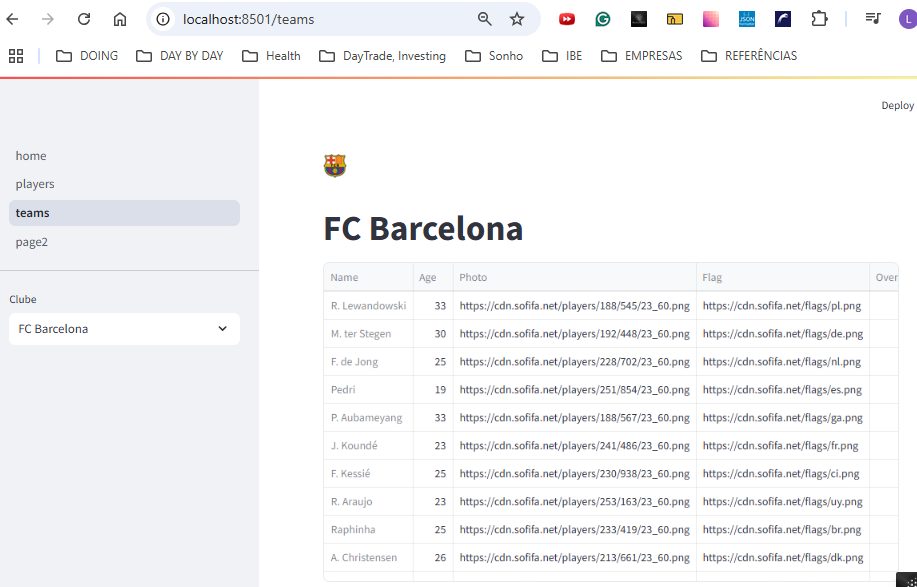
st.image(df\_filtered.iloc[0]["Club Logo"])

st.markdown(f"# {club}")

**columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']**

**st.dataframe(df\_filtered[columns])**





Vamos colocar uma progress bar junto a coluna **Overall** usando a **column\_config**

import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

st.image(df\_filtered.iloc[0]["Club Logo"])

st.markdown(f"# {club}")

columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']

**st.dataframe(df\_filtered[columns],**

***column\_config*={**

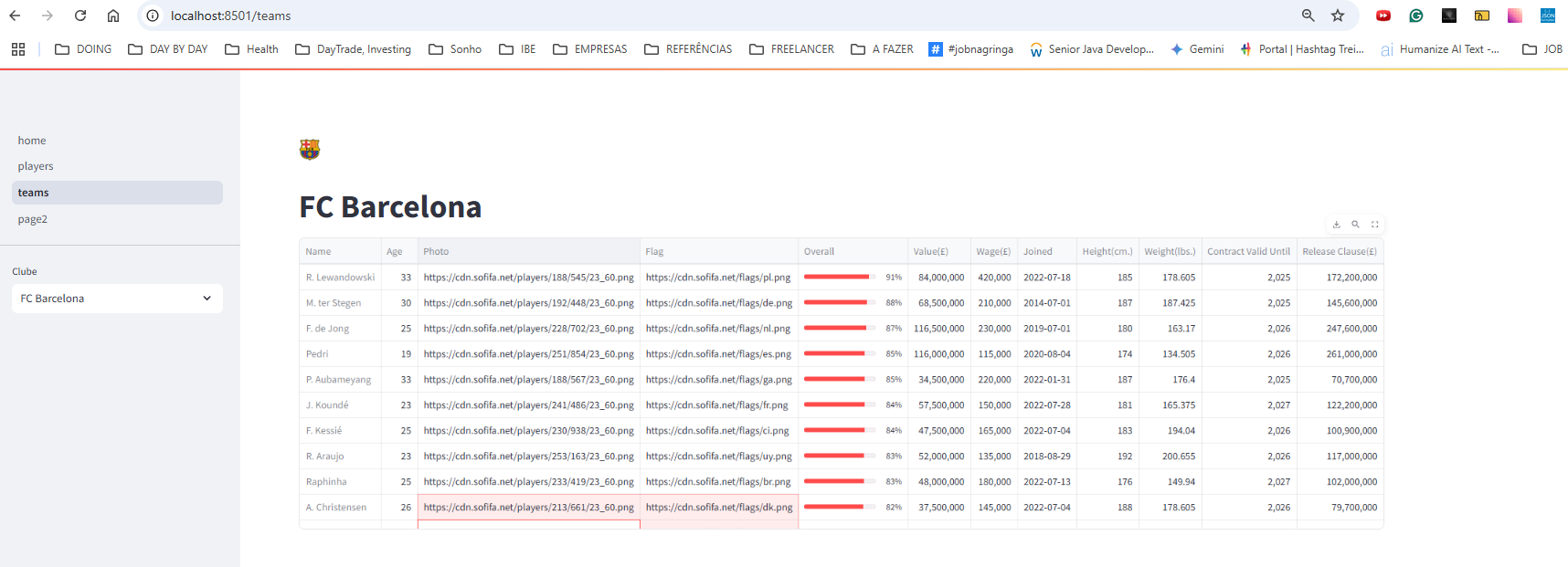
**"Overall":st.column\_config.ProgressColumn(**

**"Overall", *min\_value*=0, *max\_value*=100**

**)**

**}**

**)**

****

Vamos passar o **Wage**  como progress bar e tendo o valor máximo como o máximo do data set

import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

st.image(df\_filtered.iloc[0]["Club Logo"])

st.markdown(f"# {club}")

columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']

st.dataframe(df\_filtered[columns],

*column\_config*={

                 "Overall":st.column\_config.ProgressColumn(

                     "Overall", *min\_value*=0, *max\_value*=100

                 ),

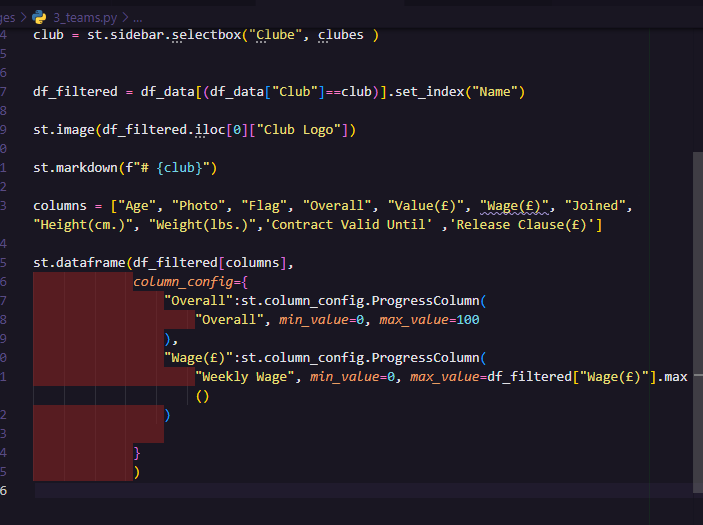
**"Wage(£)":st.column\_config.ProgressColumn(**

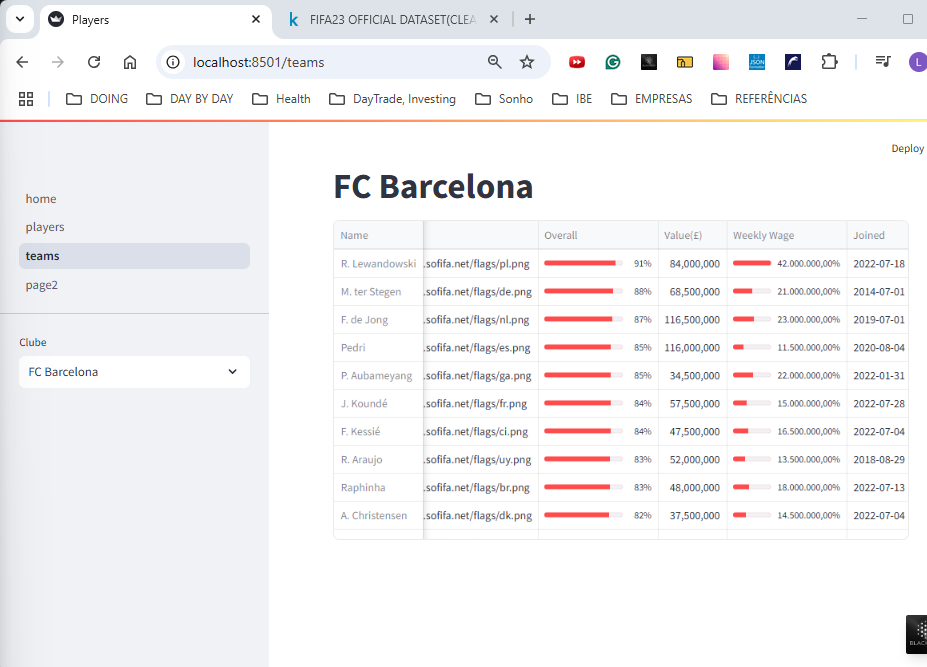
**"Weekly Wage", *min\_value*=0, *max\_value*=df\_filtered["Wage(£)"].max()**

**)**

             }

             )





import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

st.image(df\_filtered.iloc[0]["Club Logo"])

st.markdown(f"# {club}")

columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']

st.dataframe(df\_filtered[columns],

*column\_config*={

                 "Overall":st.column\_config.ProgressColumn(

                     "Overall", *min\_value*=0, *max\_value*=100

                 ),

**"Wage(£)":st.column\_config.ProgressColumn(**

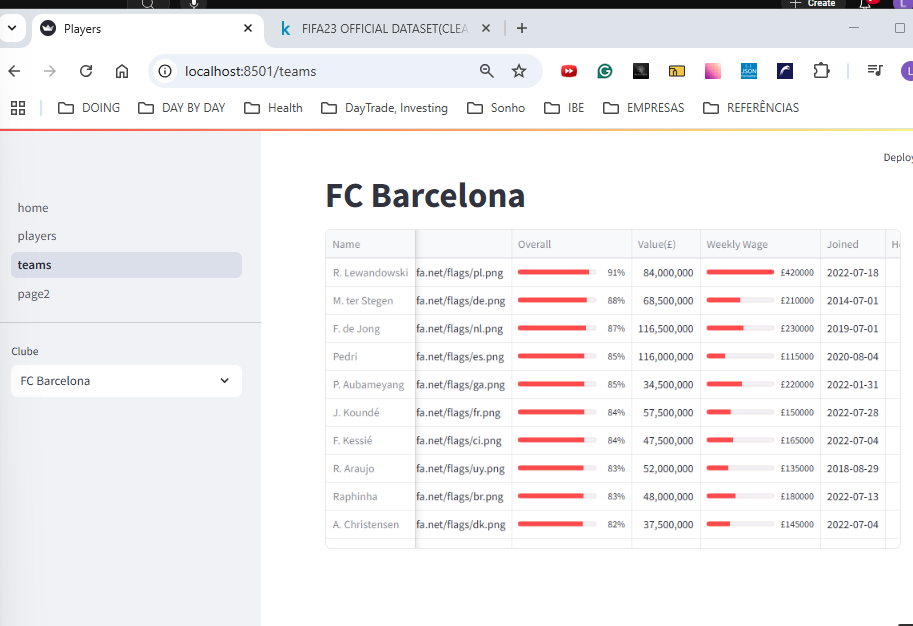
**"Weekly Wage", *format*="£%f", *min\_value*=0, *max\_value*=df\_filtered["Wage(£)"].max()**

                 )

             }

             )





Vamos colocar a photo

import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

st.image(df\_filtered.iloc[0]["Club Logo"])

st.markdown(f"# {club}")

columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']

st.dataframe(df\_filtered[columns],

*column\_config*={

                 "Overall":st.column\_config.ProgressColumn(

                     "Overall", *min\_value*=0, *max\_value*=100

                 ),

                 "Wage(£)":st.column\_config.ProgressColumn(

                     "Weekly Wage", *format*="£%f", *min\_value*=0, *max\_value*=df\_filtered["Wage(£)"].max()

                 ),

**"Photo": st.column\_config.ImageColumn()**

             }

             )



Vamos colocar a imagen da bandeira do país

import streamlit as st

st.set\_page\_config(

*page\_title*="Players",

*page\_icon*="🏃🏼",

*layout*="wide"

)

df\_data = st.session\_state["data"]

clubes  = df\_data["Club"].unique()

club = st.sidebar.selectbox("Clube", clubes )

df\_filtered = df\_data[(df\_data["Club"]==club)].set\_index("Name")

st.image(df\_filtered.iloc[0]["Club Logo"])

st.markdown(f"# {club}")

columns = ["Age", "Photo", "Flag", "Overall", "Value(£)", "Wage(£)", "Joined", "Height(cm.)", "Weight(lbs.)",'Contract Valid Until' ,'Release Clause(£)']

st.dataframe(df\_filtered[columns],

*column\_config*={

                 "Overall":st.column\_config.ProgressColumn(

                     "Overall", *min\_value*=0, *max\_value*=100

                 ),

                 "Wage(£)":st.column\_config.ProgressColumn(

                     "Weekly Wage", *format*="£%f", *min\_value*=0, *max\_value*=df\_filtered["Wage(£)"].max()

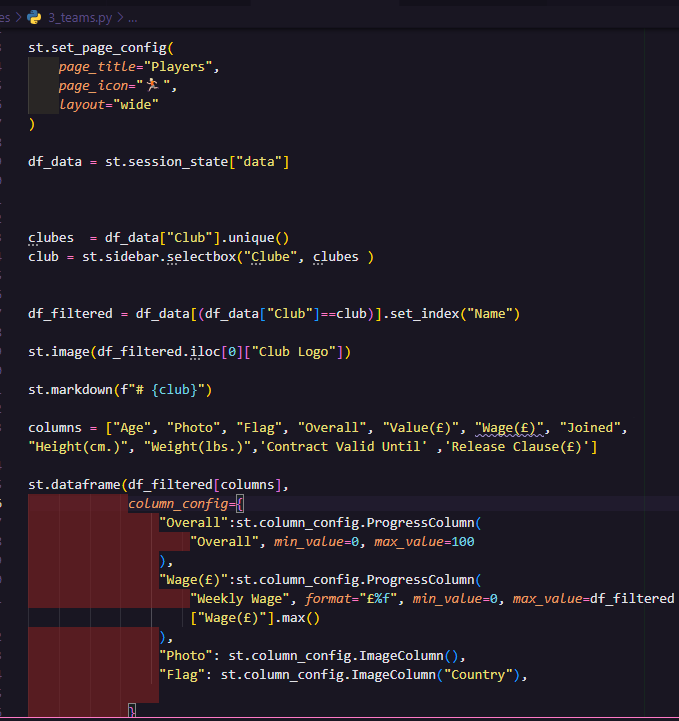
                 ),

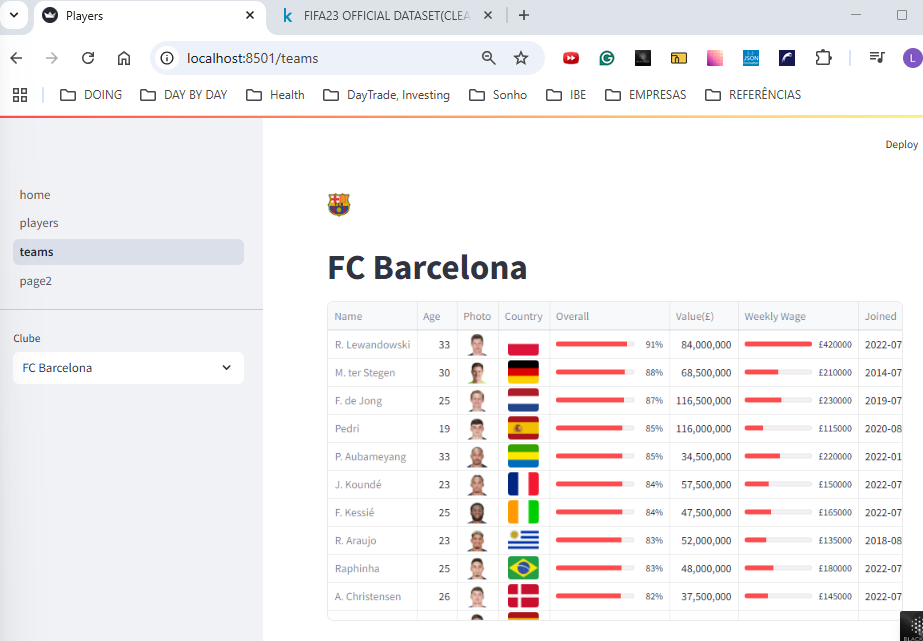
                 "Photo": st.column\_config.ImageColumn(),

**"Flag": st.column\_config.ImageColumn("Country"),**

             }

             )





## [View de jogadores e teams](https://hub.asimov.academy/curso/atividade/view-de-jogadores-e-teams/)

# Deploy com Streamlit Cloud

## [Apresentação projeto – Deploy Streamlit Cloud](https://hub.asimov.academy/curso/atividade/apresentacao-projeto-deploy-streamlit-cloud/)

## [Deploy completo no Streamlit Cloud](https://hub.asimov.academy/curso/atividade/deploy-completo-no-streamlit-cloud/)