In [ ]:

```
# PRACTICA FINAL
# CURSO: M1 PROGRAMACIÓ EN PYTHON
# DATA: 01-DEC-2022
# ALUMNE: TINO PEREZ
# TEMA: WEB SCRAPPING & WORD CLOUD

# Basado en el ejemplj del sigüiente enlace:
# https://antonio-fernandez-troyano.medium.com/nube-de-palabras-word-cloud-con-pythor
# y modificado para mostrar la frecuencia con la que aparecen los nombres de las sele
# en el Mundial Qatar 2022 en los 5 principales diarios deportivos de España.
# Se ha eliminado el código y funciones no usadas
# Los resultados se grafican en un wordcloud clasico y en dos wordclod de contorno (F
# la ventana

# TO DO: presentar los gráficos con tkinter

#Librerías básicas utilizadas
import numpy as np
import pandas as pd
import tkinter
from tkinter import *

#Librerías necesarias para scrapear textos de páginas web
import requests
from bs4 import BeautifulSoup

#Librerías necesarias para abrir imágenes, generar nube de palabras y visualizar imág
from PIL import Image, ImageTk
from wordcloud import WordCloud, ImageColorGenerator

#Función para transformar todas las imágenes PNG con fondo transparente a fondo blanc

def transform_white_backgroud(png_path):
    picture = Image.open(png_path).convert("RGBA")
    image = Image.new("RGB", picture.size, "WHITE")
    image.paste(picture, (0, 0), picture)
    mask = np.array(image)

    return mask

# Funcion para scrappear las webs en busca de las selecciones
def get_texto_url(list_webs):

    # Hay que incluir unos "headers" para que las páginas web piensen que es una pers
    # a la web
    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
                   "Accept-Encoding": "*", "Connection": "keep-alive"
           }

    texto =" "
    long=0
    for url in webs:
        # Para cada una de las webs definidas
        try:
            web = requests.get(url, headers=headers)

            soup = BeautifulSoup(web.text)
```

```
            # Creamos el texto con las selecciones (teams) encontradas en la pagina w
            for i,n in enumerate(teams):
                parrafos = soup.find_all(text=n,recursive = True)
                for p in parrafos:
                    long += len(p.text)
                    texto += "|"+ str(p.text)    # Añadimos el separador | por cada p
                                                 # con nombre compuesto y añada como
        # Gestión de errores
        except:
            print("Error en la url {}, response {}".format(url,web))

    #print("Se han añadido un total de {} caractéres a la variable texto".format(long
    #print(texto)    # debug
    return texto

# Funcion para crear un objeto photoimage de la imagen pasada en el path en la posici
def show_img (img,pos_x, pos_y):
    image1 = Image.open(img)

    test = ImageTk.PhotoImage(image1)

    label1 = tkinter.Label(image=test)
    label1.image = test

    # Posiciona la imagen
    label1.place(x=int(pos_x), y=int(pos_y))

# Lista de webs a scrapear
webs = ['https://www.mundodeportivo.com',
        'https://www.marca.com',
        'https://as.com',
        'https://www.sport.es',
        'https://www.donbalon.com']

# Lista de selecciones participantes
teams = ['Alemania', 'Arabia Saudí', 'Argentina', 'Australia',
         'Bélgica', 'Brasil', 'Camerún', 'Canadá',
         'Qatar', 'Corea del Sur', 'Costa Rica', 'Croacia',
         'Dinamarca', 'Ecuador', 'España', 'Estados Unidos',
         'Francia', 'Gales', 'Ghana', 'Inglaterra',
         'Irán', 'Japón', 'Marruecos', 'México',
         'Países Bajos', 'Polonia', 'Portugal', 'Senegal',
         'Serbia', 'Suiza', 'Túnez', 'Uruguay'
        ]

texto = get_texto_url(webs)

#Creamos una lista con todos los paises delimitados por |
lista_texto = texto.split("|")
#print(lista_texto)    # debug
paises = []
for pais in lista_texto:
    paises.append(pais)

#Generamos un diccionario para contabilizar los paises:
word_count={}

for pais in paises:
    if pais in word_count.keys():
```

```
            word_count[pais][0]+=1
        else:
            word_count[pais]=[1]

    #print(word_count)    # debug

    #Generamos el DataFrame y lo ordenamos:

    df = pd.DataFrame.from_dict(word_count).transpose()
    df.columns=["freq"]
    df.sort_values(["freq"], ascending=False, inplace=True)
    #print(df)    #debug
    sorted_txt = ','.join(df.index.values)
    sorted_text = sorted_txt.replace(" ", "") # unimos los nombre de los paises con nombr
                                              # ya que wordcloud cuenta los paises como d
    print(sorted_text)    # debug

    #WordCloud sencillo

    word_cloud = WordCloud(height=800, width=800, background_color='white',max_words=150,
                           collocations=False, collocation_threshold=30).generate(sorted_

    word_cloud.to_file("./img/ejemplo_sencillo.png") #Guardamos la imagen generada

    # crear un objeto photoimage para el word cloud generado

    root = Tk()
    root.geometry('800x800')   # Medida de la ventana
    root.resizable(0,0)        # Desactivar redimensión de ventana (1,1 para activar)
    root.title("WordCloud Sencillo")

    show_img("./img/ejemplo_sencillo.png", 10, 10)    # Llamada a la funcion

    root.mainloop()

    #Word cloud aplicando una máscara de contorno Camel y los colores de la máscara

    mask = transform_white_backgroud("./img/camel.jpg")

    image_colors = ImageColorGenerator(mask) #Generamos los colores de la propia máscara

    word_cloud = WordCloud(mask=mask, background_color='white', contour_width=1, contour_
                           max_words=150, min_font_size=5, collocations=False, collocatio

    word_cloud.to_file("./img/camel_color.png") #Guardamos la imagen generada

    # crear un objeto photoimage para el word cloud generado

    root1 = Tk()
    root1.geometry('800x800')   # Medida de la ventana
    root1.resizable(0,0)        # Desactivar redimensión de ventana (1,1 para activar)
    root1.title("WordCloud en Camel")

    show_img("./img/camel_color.png", 0, 0)    # Llamada a la funcion

    root1.mainloop()

    #Word cloud aplicando una máscara de contorno Python y los colores de la máscara

    mask = transform_white_backgroud("./img/python.png")
```

```
mask = transform_white_background( './img/python.png' )

image_colors = ImageColorGenerator(mask) #Generamos los colores de la propia máscara

word_cloud = WordCloud(mask=mask, background_color='white', contour_width=1, contour_
                       max_words=150, min_font_size=5, collocations=False, collocatio

word_cloud.to_file("./img/python_color.png") #Guardamos la imagen generada DESPUÉS DE

# crear un objeto photoimage para el word cloud generado

root2 = Tk()
root2.geometry('800x800')   # Medida de la ventana
root2.resizable(0,0)        # Desactivar redimensión de ventana (1,1 para activar)
root2.title("WordCloud en Python")

show_img("./img/python_color.png", 0, 0)    # Llamada a la funcion

root2.mainloop()
```

```
Argentina,España,Francia,Japón,Alemania,Canadá,Marruecos,CostaRica,Croacia,Bélgica,Au
stralia,México,Qatar,Túnez,Brasil,EstadosUnidos,Portugal,Polonia,ArabiaSaudí,PaísesBa
jos,Suiza,Serbia,Senegal,Gales,Irán,Inglaterra,Ghana,Ecuador,Dinamarca,Camerún,Urugua
y,,CoreadelSur
```

In [ ]: