Pràctica Individual :

Crea un projecte Python que funcioni i es pugui presentar als teus companys. Ha de tenir més de 50 línies de codi.

Has de lliurar una documentació amb els següents apartats, a més del programa.

Nom i cognoms de l'alumne

Oriac Gimeno Lozano

Nom del Projecte

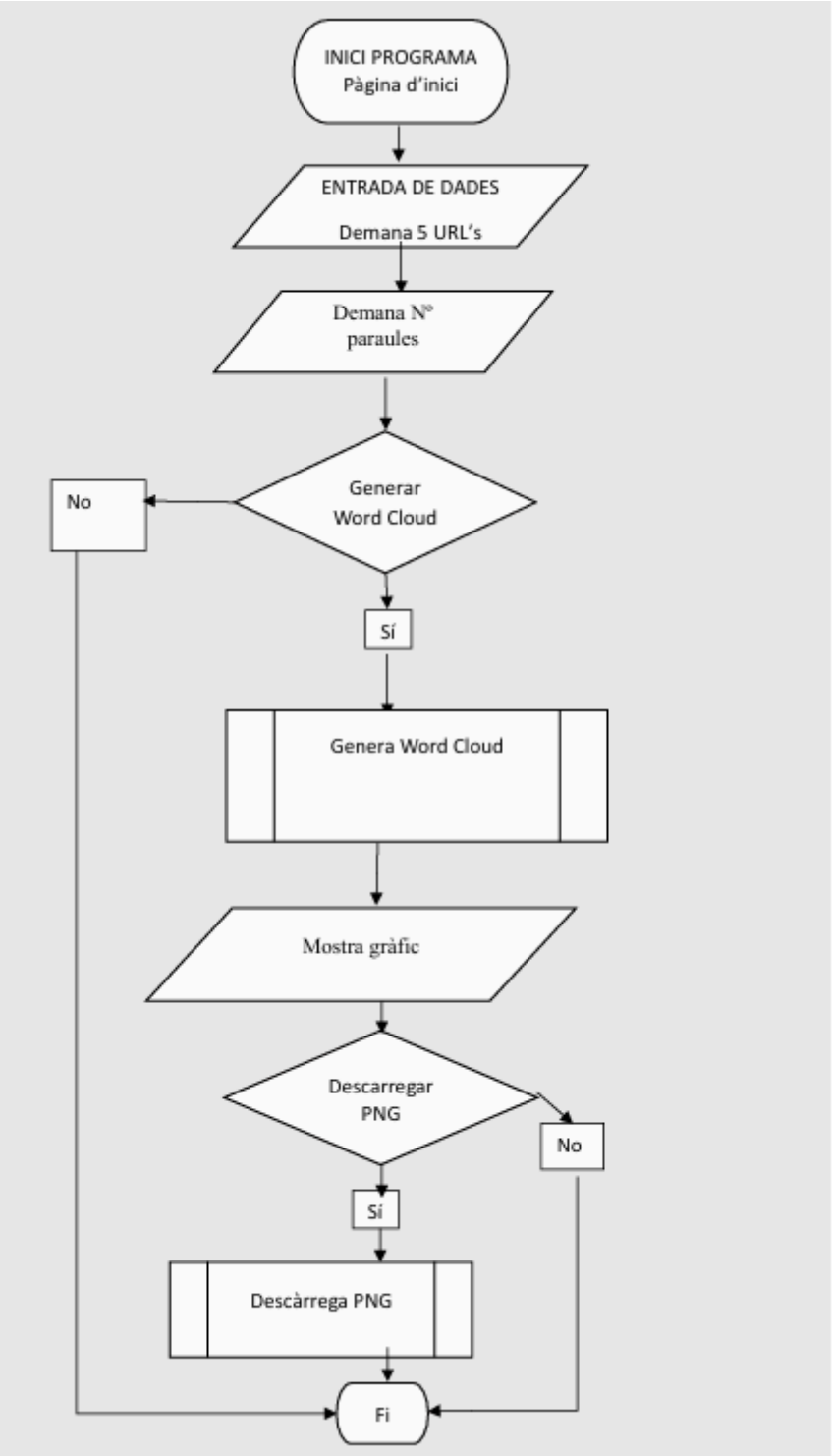
News Word Cloud

Descripció

És una página web feta amb Streamlit on l'usuari pot introduir fins a 5 pàgines web de portals de notícies, diaris, etc.. en castellà, i l'aplicació elabora i mostra un WordCloud amb el número de paraules prèviament introduïdes per l'usuari i mostra les paraules més significatives i esmentades entre les webs analitzades. Tot seguit permet descarregar la imatge generada en un arxiu .PNG.

Esbós de la pantalla (o wireframe)

Diagrama de flux



Codi del programa

```
import streamlit as st #Per a crear la interficie web amb Streamlit
from wordcloud import WordCloud, STOPWORDS #generar el núvol de paraules i incloure paraules comuns a ignorar (stopwords)
import matplotlib.pyplot as plt #Per a per mostrar el WordCloud
import requests #Per a  per fer peticions HTTP i obtenir el contingut de les URLs
from bs4 import BeautifulSoup, Comment # Llibreria per extreure i parsejar dades HTML
from io import BytesIO #Per a la generació i descàrrega d'imatges en format de flux de bytes
import base64 #Per a  convertir la imatge del WordCloud a base64 per descarregar-la
import re #Per a fer expressions regulars, com eliminar parts no desitjades d'un text

# Interface de Streamlit
st.title("Haz tu WordCloud con tus webs favoritas")



# Entrada de l'usuari per a les URLs
urls = []
for i in range(5):
    url = st.text_input(f"Introduce la URL {i+1}:", "")
    if url:
        urls.append(url)



# Funció per a extreure el text de les URLs
def get_text_from_url(url):

#S'ha d'incloure uns "headers" per a que les pàgines pensin que es una persona "real" qui està accedint a la web
    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.121 Safari/537.36",
               "Accept-Encoding": "*", "Connection": "keep-alive"
```

```
        }
    try:
        response = requests.get(url, headers=headers)
        soup = BeautifulSoup(response.content, 'html.parser')

        # Extreure text de paràgrafs
        paragraphs = ' '.join([p.text for p in soup.find_all('p')])



        # Combinar el text dels paràgrafs amb els headers ponderats
        return paragraphs + ' '
    except Exception as e:
        st.error(f"No s'ha pogut obtenir el text de la URL: {url}. Error: {e}")
        return ""

# Demanar a l'usuari que introdueixi el nombre de paraules pel WordCloud
num_paraules = st.number_input("Indica el nombre de palabres para tu WordCloud:", min_value=1, max_value=500)

paraules_excloure = st.text_input("Introdueix paraules per excloure (separades per comes):")
# Camp de text per a les paraules a afegir a stop_words

paraules_excloure = [paraula.strip() for paraula in paraules_excloure.split(",") if paraula.strip()]

# Afegir les noves paraules a `stop_words_personalitzades`
custom_stopwords.update(paraules_excloure)



# Funció per generar el WordCloud a partir del text amb la quantitat assignada a num_paraules
def generate_wordcloud(text, custom_stopwords, num_paraules):

    wordcloud = WordCloud(
        width=800, height=500, background_color="white", #Mida del WordCloud i color de fons
        stopwords=custom_stopwords, max_words=num_paraules
    ).generate(text)
```

```
    return wordcloud



 # Funció per descarregar el WordCloud
def download_wordcloud(image):
    buf = BytesIO()
    image.savefig(buf, format="png")
    buf.seek(0)
    image_base64 = base64.b64encode(buf.read()).decode()
    href = f'<a href="{image_base64}"
download="wordcloud.png">Descarrega el WordCloud</a>'
    return href

# Llista de preposicions i monosíl·labs a eliminar
preposiciones = {
    "a", "ante", "bajo", "cabe", "con", "contra", "de", "desde", "durante", "en", "entre",
    "hacia", "hasta", "mediante", "para", "por", "según", "sin", "so", "sobre", "tras",
"versus", "vía"
}

monosilabos = {
    "a", "al", "as", "da", "de", "di", "do", "el", "en", "es", "ha", "he", "la", "le", "lo", "me",
"mi",
    "ni", "no", "os", "se", "si", "su", "ta", "te", "tu", "un", "va", "ya", "yo"
}
# Llista d'adverbis
adverbios = {
        'poco', 'demasiado', 'nada', 'mucho', 'todo', 'nada', 'algo','bien', 'mal', 'mejor',
'peor', 'deprisa', 'despacio', 'cómodamente','antes', 'ahora', 'después', 'hoy', 'mañana',
'luego', 'todavía','aquí', 'ahí', 'allá', 'cerca', 'lejos', 'dentro', 'fuera', 'alrededor','sí', 'no',
'también', 'claro', 'cierto', 'efectivamente','no', 'nada', 'tampoco', 'jamás',
'nunca','quizá','quizás', 'tal', 'vez', 'acaso', 'a lo  mejor', 'probablemente'
}

# Paraules poc rellevants a eliminar
palabras_irrelevantes = {
    "comentarios", "comentario", "opiniones", "opinion", "artículo", "articulo",
    "tema", "temas", "sección", "secciones", "noticia", "noticias"
}
```

```
#Afegim alguns caràcters que hem trobat
palabras_anadidas = {"\u200b", "\xa0",
"editorial","Lectores","OFRECIDO","puede","cómo","hacer","forma","parte","además",
"según","tanto","algo","dicho","sido"}

stop_words = ['de', 'la', 'que', 'el', 'en', 'y', 'a', 'los', 'del', 'se', 'las', 'por', 'un', 'para',
'con', 'no', 'una', 'su', 'al', 'lo', 'como', 'más', 'pero', 'sus', 'le', 'ya', 'o', 'este', 'sí',
'porque', 'esta', 'entre', 'cuando', 'muy', 'sin', 'sobre', 'también', 'me', 'hasta', 'hay',
'donde', 'quien', 'desde', 'todo', 'nos', 'durante', 'todos', 'uno', 'les', 'ni', 'contra', 'otros',
'ese', 'eso', 'ante', 'ellos', 'e', 'esto', 'mí', 'antes', 'algunos', 'qué', 'unos', 'yo', 'otro',
'otras', 'otra', 'él', 'tanto', 'esa', 'estos', 'mucho', 'quienes', 'nada', 'muchos', 'cual',
'poco', 'ella', 'estar', 'estas', 'algunas', 'algo', 'nosotros', 'mi', 'mis', 'tú', 'te', 'ti', 'tu', 'tus',
'ellas', 'nosotras', 'vosotros', 'vosotras', 'os', 'mío', 'mía', 'míos', 'mías', 'tuyo', 'tuya',
'tuyos', 'tuyas', 'suyo', 'suya', 'suyos', 'suyas', 'nuestro', 'nuestra', 'nuestros', 'nuestras',
'vuestro', 'vuestra', 'vuestros', 'vuestras', 'esos', 'esas', 'estoy', 'estás', 'está', 'estamos',
'estáis', 'están', 'esté', 'estés', 'estemos', 'estéis', 'estén', 'estaré', 'estarás', 'estará',
'estaremos', 'estaréis', 'estarán', 'estaría', 'estarías', 'estaríamos', 'estaríais', 'estarían',
'estaba', 'estabas', 'estábamos', 'estabais', 'estaban', 'estuve', 'estuviste', 'estuvo',
'estuvimos', 'estuvisteis', 'estuvieron', 'estuviera', 'estuvieras', 'estuviéramos',
'estuvierais', 'estuvieran', 'estuviese', 'estuvieses', 'estuviésemos', 'estuvieseis',
'estuviesen', 'estando', 'estado', 'estada', 'estados', 'estadas', 'estad', 'he', 'has', 'ha',
'hemos', 'habéis', 'han', 'haya', 'hayas', 'hayamos', 'hayáis', 'hayan', 'habré', 'habrás',
'habrá', 'habremos', 'habréis', 'habrán', 'habría', 'habrías', 'habríamos', 'habríais',
'habrían', 'había', 'habías', 'habíamos', 'habíais', 'habían', 'hube', 'hubiste', 'hubo',
'hubimos', 'hubisteis', 'hubieron', 'hubiera', 'hubieras', 'hubiéramos', 'hubierais',
'hubieran', 'hubiese', 'hubieses', 'hubiésemos', 'hubieseis', 'hubiesen', 'habiendo',
'habido', 'habida', 'habidos', 'habidas', 'soy', 'eres', 'es', 'somos', 'sois', 'son', 'sea',
'seas', 'seamos', 'seáis', 'sean', 'seré', 'serás', 'será', 'seremos', 'seréis', 'serán', 'sería',
'serías', 'seríamos', 'seríais', 'serían', 'era', 'eras', 'éramos', 'erais', 'eran', 'fui', 'fuiste',
'fue', 'fuimos', 'fuisteis', 'fueron', 'fuera', 'fueras', 'fuéramos', 'fuerais', 'fueran', 'fuese',
'fueses', 'fuésemos', 'fueseis', 'fuesen', 'sintiendo', 'sentido', 'sentida', 'sentidos',
'sentidas', 'siente', 'sentid', 'tengo', 'tienes', 'tiene', 'tenemos', 'tenéis', 'tienen', 'tenga',
'tengas', 'tengamos', 'tengáis', 'tengan', 'tendré', 'tendrás', 'tendrá', 'tendremos',
'tendréis', 'tendrán', 'tendría', 'tendrías', 'tendríamos', 'tendríais', 'tendrían', 'tenía',
'tenías', 'teníamos', 'teníais', 'tenían', 'tuve', 'tuviste', 'tuvo', 'tuvimos', 'tuvisteis',
'tuvieron', 'tuviera', 'tuvieras', 'tuviéramos', 'tuvierais', 'tuvieran', 'tuviese', 'tuvieses',
'tuviésemos', 'tuvieseis', 'tuviesen', 'teniendo', 'tenido', 'tenida', 'tenidos', 'tenidas',
'tened']

# Funció per filtrar les paraules de 3 lletres
def filter_three_letter_words(text):
    return ' '.join([word for word in text.split() if len(word) > 3])
```

```
# Funció per eliminar paraules poc rellevants
def remove_irrelevant_words(text, url):
    # Elimina el nom de la URL
    url_name = re.sub(r'https?://(www\.)?', '', url)  # Elimina el protocol i 'www'
    url_name = re.sub(r'/.*$', '', url_name)  # Manté només el domini

    # Combina les paraules que volem eliminar
    words_to_remove = set(url_name.split()).union(palabras_irrelevantes)

    return ' '.join([word for word in text.split() if word not in words_to_remove])

 #-.---------------------------------------------- inici
# Combina les stop-words amb les preposicions, monosíl·labs, paraules poc rellevants i paraules afegides

custom_stopwords = set(stop_words).union(preposiciones).union(monosilabos).union(palabras_irrelevantes).union(palabras_anadidas).union(adverbios)



# Botó per generar el WordCloud
if st.button("Generar WordCloud"):
    all_text = ""
    for url in urls:
        text = get_text_from_url(url)
        # Eliminar paraules poc rellevants
        cleaned_text = remove_irrelevant_words(text, url)
        all_text += cleaned_text

    if all_text:
        # Filtrar paraules de 3 lletres
        filtered_text = filter_three_letter_words(all_text)



        wordcloud = generate_wordcloud(filtered_text, custom_stopwords, num_paraules)

        # Mostrar el WordCloud
```

```
fig, ax = plt.subplots()
ax.imshow(wordcloud, interpolation='bilinear')
ax.axis("off")
st.pyplot(fig)

# Oferir la descàrrega
st.markdown(download_wordcloud(fig), unsafe_allow_html=True)
```

Exemples de resultats del programa