

UNIVERSIDAD DE GUAYAQUIL
Facultad de Ciencias Matematicas y Fisicas
Carrera de Software
Sof-Ma-3-1
Proyecto - Aplicación de Modelo de Desarrollo

Marcillo Falcones Fernando
Matamoros Jalca Joaquin
Vera Lenis Bryan

16 de agosto de 2020

Índice

1. MODELO DE DESARROLLO POR PROTOTIPOS RAPIDOS	3
1.1. Recolección y Refinamiento de Requisitos	3
2. Diseño Rapido y Modelado	3
2.1. Uso de Beautiful Soup	3
2.2. Librería Requests	4
2.3. Primer diseño de generación de nube de palabras	4
3. CONSTRUCCION DEL PROTOTIPO	5
3.1. Evaluación	5
4. DESARROLLO, ENTREGA Y RETROALIMENTACION	6
4.1. Conclusiones y Entrega Final	7

1. MODELO DE DESARROLLO POR PROTOTIPOS RAPIDOS

1.1. Recolección y Refinamiento de Requisitos

El proyecto tiene como objetivo generar una nube de palabras de etiquetas por usuario de la plataforma Stack Overflow en español. Para ello, el usuario debe ingresar el ID del Usuario de Stack Overflow en español, y la aplicación debe ser capaz de generar su nube de palabras (imagen) de etiquetas. REQUERIMIENTOS: Lenguaje de programación: PYTHON Ingresar y leer el código de usuario con el fin de obtener así su etiqueta. Generar un archivo txt para almacenar las etiquetas. Leer el archivo tipo txt y generar una nube de palabras con las etiquetas en un .JPG. Tiempo: 2 semanas

2. Diseño Rápido y Modelado

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final.

Primer diseño de extracción de usuarios de StackOverflow.

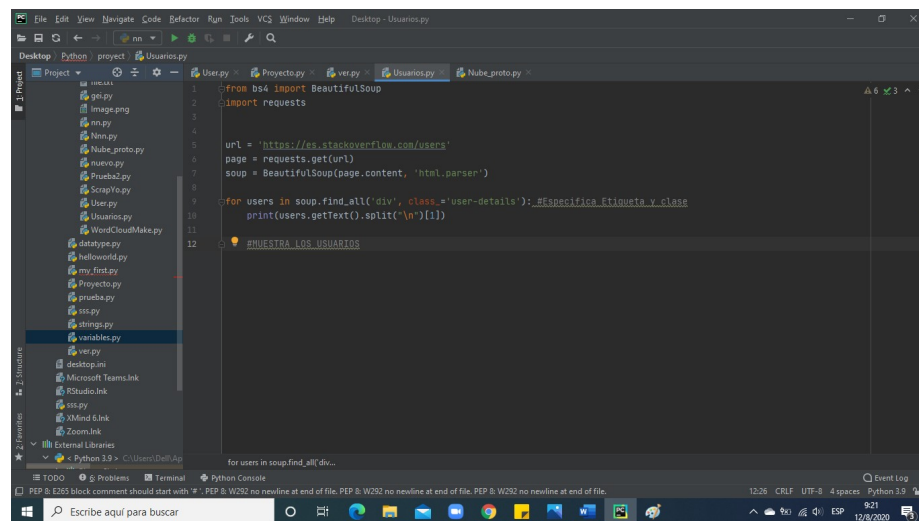


Ilustración 1. Diseño Rápido

2.1. Uso de BeautifulSoup

Beautiful Soup es una librería de Python para analizar documentos HTML (incluyendo los que tienen un marcado incorrecto). Esta librería crea un árbol con todos los elementos del documento y puede ser utilizado para extraer información. Por lo tanto, esta librería es útil para hacer web scraping (extraer

información de sitios web) (Richardson, s.f.)

2.2. Librería Requests

La librería requests la utilizaremos para realizar las peticiones a la página de la que vamos a extraer los datos.

2.3. Primer diseño de generación de nube de palabras

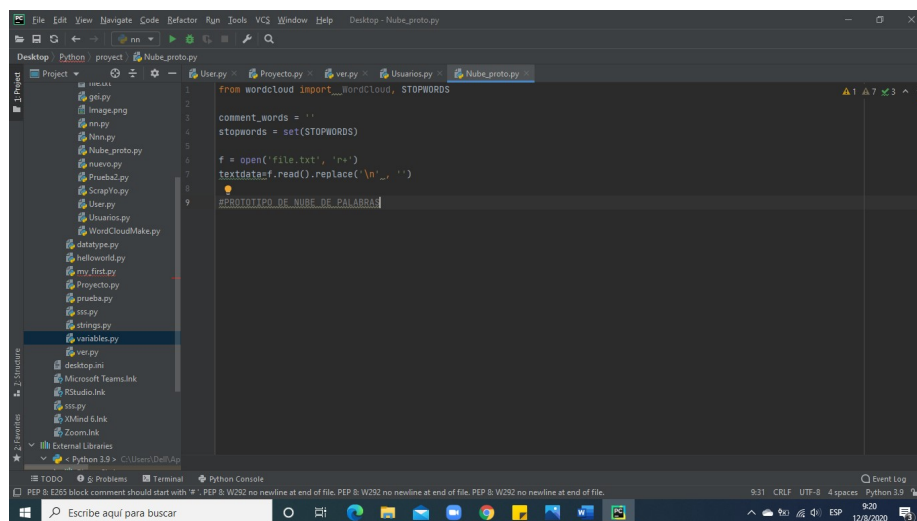


Ilustración 2. Primer Diseño de Generación de Nube de Palabras

Word Cloud es una técnica de visualización de datos que se utiliza para representar datos de texto en los que el tamaño de cada palabra indica su frecuencia o importancia. Los puntos de datos de texto importantes se pueden resaltar usando una nube de palabras. Las nubes de palabras se utilizan ampliamente para analizar datos de sitios web de redes sociales. Para generar nubes de palabras en Python, los módulos necesarios son: matplotlib, pandas y wordcloud. (Python, s.f.)

3. CONSTRUCCION DEL PROTOTIPO

Como estamos usando Modelo de Desarrollo de Prototipos rápido, nos desviamos hacia el uso de la librería Scrapy, la cual es una librería que permite hacer web scraping de manera vertical y horizontal, es decir, vamos a acceder a todas las páginas web donde se encuentran los usuarios de StackOverflow.



Paginación.

3.1. Evaluación

Después de varias pruebas se llegó a una serie de conclusiones y nos encontramos con los siguientes problemas: No se puede hacer demasiado requerimientos ya que nos lleva a un baneo temporal o permanente de la IP Extrae la información incompleta

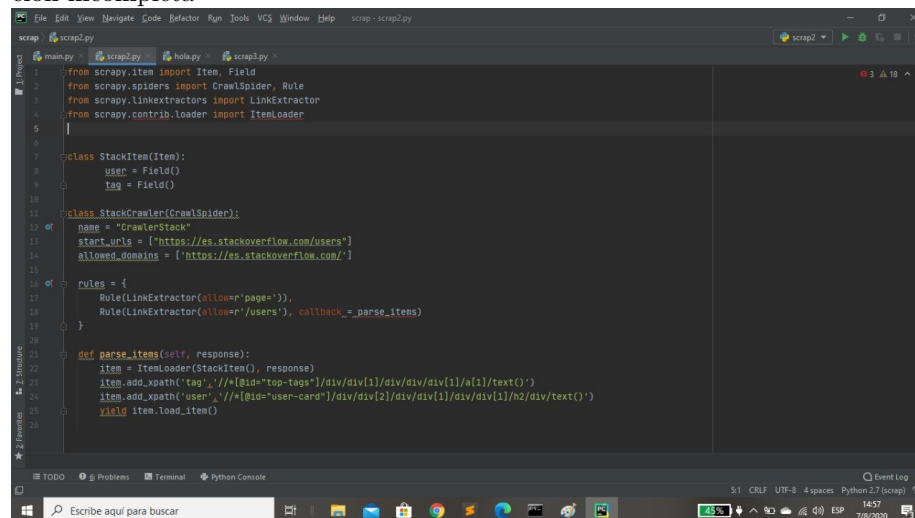
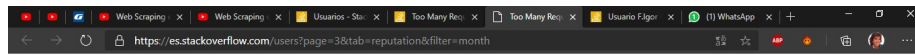


Ilustración 3. Baneo de Ip



Too many requests

This IP address (45.71.115.106) has performed an unusual high number of requests and has been temporarily rate limited. If you believe this to be in error, please contact us at team@stackexchange.com.

When contacting us, please include the following information in the email:

Method: rate limit

XID: 1011621485-MIA

IP: 45.71.115.106

X-Forwarded-For: 45.71.115.106

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36;KHTML, like Gecko; Chrome/84.0.4147.105 Safari/537.36 Edg/84.0.522.56)

Reason: Request rate

Time: Tue, 11 Aug 2020 19:35:33 GMT

URL: es.stackoverflow.com/users?page_3_tab_reputation_filter_month

Browser Location: https://es.stackoverflow.com/users?page=3&tab=reputation&filter=month

Ilustración 4. Baneo de Ip

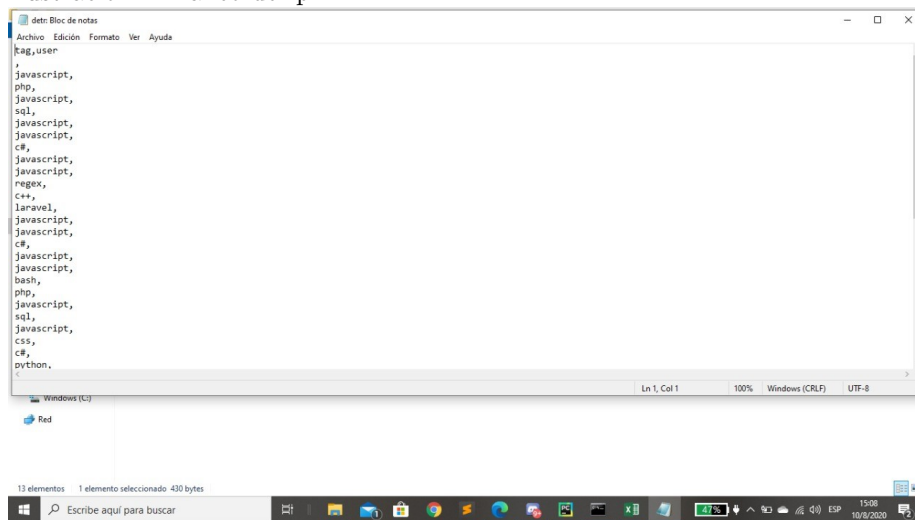


Ilustración 5. Informacion Completa

4. DESARROLLO, ENTREGA Y RETROALIMENTACION

Haciendo una retroalimentación entre todos los integrantes del grupo, se llegó a una conclusión, y mediante varios errores nos dimos cuenta de lo siguiente: SI copias la URL de un usuario, y la pegas en otra ventana, es suficiente con que cumpla la siguiente sintaxis para que funcione:
url = url + user + &tab=tags”

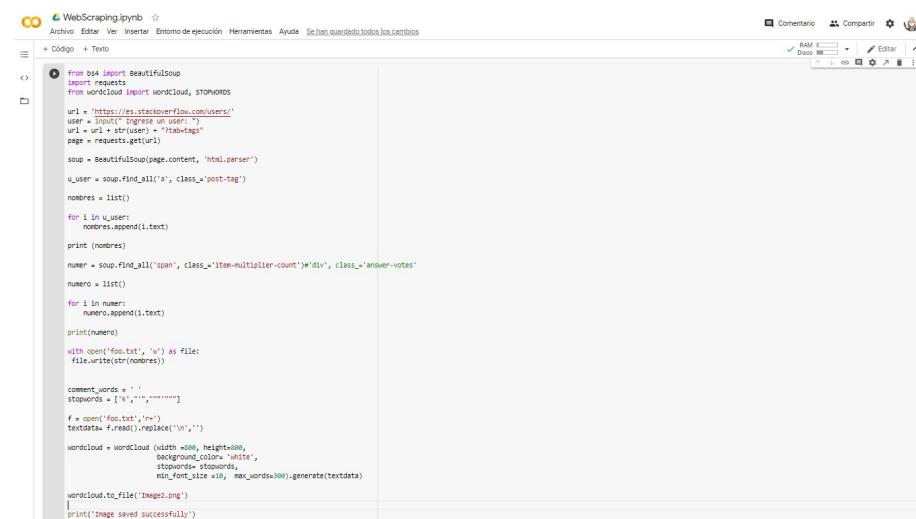
Lo cual en código nos quedó así:

```
url = url + str(user) + "&tab=tags"
```

Es decir, cuando le damos un user correcto, el programa hace una extracción de datos única, con lo cual optimizamos tiempo y espacio de memoria, además de que obtendremos siempre información actualizada. Así que descartamos por completo el segundo prototipo y regresamos al primer programa, modificando cierta parte de su estructura, además concatenando este con el de la nube, la cual recibe un archivo .txt y genera la nube sin ningún inconveniente.

4.1. Conclusiones y Entrega Final

En cada fase de nuestro trabajo obtuvimos retroalimentación y analizamos ambos prototipos, y aunque se haya perdido tiempo en el desarrollo del segundo, nos sirvió para hallar un factor importante en el producto final



```
from bs4 import BeautifulSoup
import requests
from wordcloud import WordCloud, STOPWORDS

url = 'https://es.stackoverflow.com/users/'
user = input("Ingrese un user: ")
url = url + str(user) + "/?tab=tags"
page = requests.get(url)

soup = BeautifulSoup(page.content, 'html.parser')
u_user = soup.find_all('a', class_='post-tag')
nombres = list()

for i in u_user:
    nombres.append(i.text)

print(nombres)

numer = soup.find_all('span', class_='item-multiplier-count', class_='answer-votes')
numero = list()

for i in numer:
    numero.append(i.text)

print(numero)

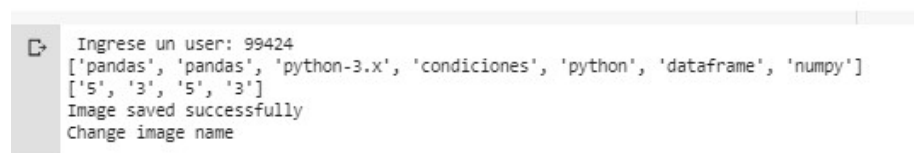
with open('foo.txt', 'w') as file:
    file.write(str(nombres))

comment_words = ''
stopwords = ['&', '!', '...', '']
f = open('foo.txt', 'r+')
textdata = f.read().replace("\n", " ")

wordcloud = WordCloud(width=800, height=800,
                        background_color='white',
                        stopwords=stopwords,
                        min_font_size=10, max_words=100).generate(textdata)

wordcloud.to_file('image.png')
print("Image saved successfully")
```

Ilustración 6. Código final en el entorno Google Colab



```
Ingrese un user: 99424
['pandas', 'pandas', 'python-3.x', 'condiciones', 'python', 'dataframe', 'numpy']
['5', '3', '5', '3']
Image saved successfully
Change image name
```

Ilustración 7. Programa ejecutado en Google Colab

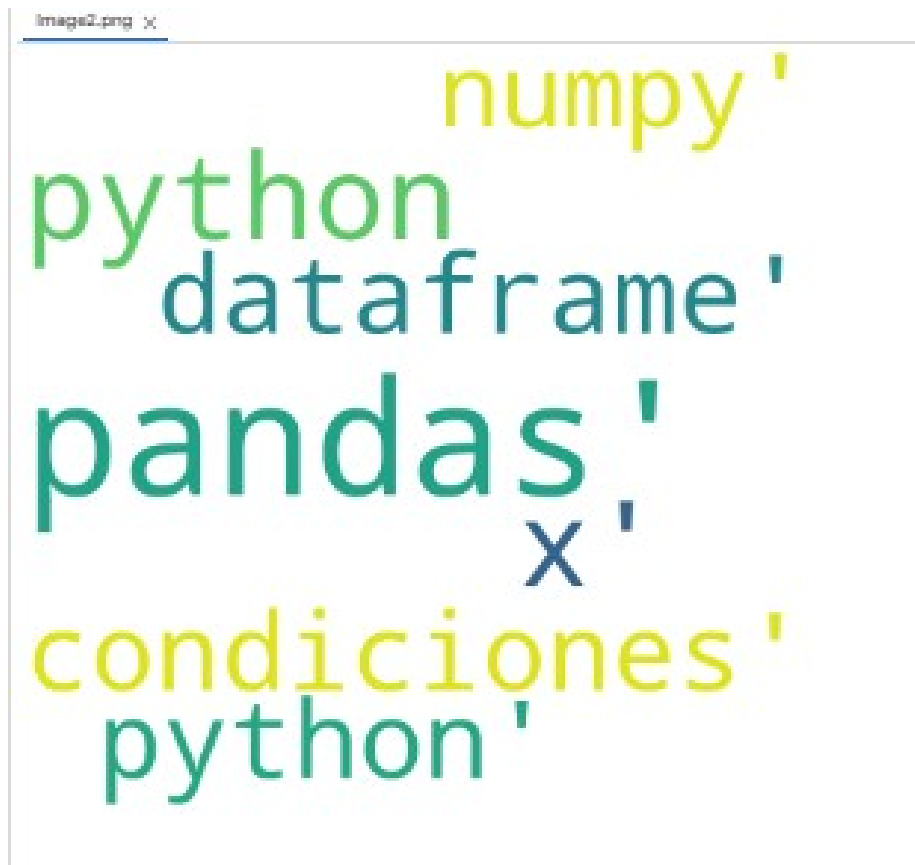


Ilustración 8. Imagen generada

GITHUB:

<https://github.com/alexisevergarden/wordcloud>

ENLACE A GOOGLE COLAB:

<https://colab.research.google.com/drive/1Wq-bEGcdJAxfekieAnYrwIg76L3-a-Eb?usp=sharing>