HOJA DE EJERCICIOS

CREACIÓN DE UN MOTOR DE BÚSQUEDA ESTÁNDAR

Se propone un ejercicio para el desarrollo de un **motor de búsqueda web** *o* ***web crawler***.

1. El motor de búsqueda debe tomar como origen una página web (esta se puede incluir en el código fuente o ser pasada como argumento al programa). A partir de esta web (se recomienda*: https://es.wikipedia.org/wiki/Wikipedia:Portada*) se deben visitar los links proporcionados por la misma. De manera recursiva se debe hacer lo mismo con las nuevas webs que se vayan analizando. Para llevar a cabo la búsqueda de los links se debe utilizar la librería de **Python *BeautifulSoup*.**
   Las páginas web que se vaya visitando deben ser almacenadas en un archivo introduciendo su URL y su texto (código HTML) concatenados como contenido. El nombre del archivo es a discreción del desarrollador.
   Se recomienda utilizar hilos de procesamiento y una cola de almacenamiento de links a visitar de tamaño 100. De esta manera el motor de búsqueda podrá crear un hilo cada poco tiempo que se encargue de realizar el procesamiento de un único link.
   El proceso de trabajo es el siguiente:
   Repetir para siempre:
   A) El programa principal extrae de la cola un link anteriormente almacenado.
   B) El programa principal crea y lanza un nuevo hilo y duerme medio segundo.
   C) El hilo procesa la URL con BeautifulSoup y extrae nuevos links.
   D) El hilo almacena los links en la cola.
   E) El hilo finaliza su actividad mientras el motor continúa iterando (automático).

   Ayuda:

   Librerías básicas para la implementación de esta herramienta:

```
from bs4 import BeautifulSoup
import _thread, queue, time, requests
```

   Para crear un hilo:

```
_thread.start_new_thread(<método a ejecutar>)
```

   Para dormir el proceso principal medio segundo mientras se ejecuta un hilo:

```
time.sleep(0.5)
```

   Para crear una cola de 100:

```
q = queue.Queue(100)
```

   Para añadir una URL a la cola:

```
q.put(new_url)
```

   Para extraer una URL de la cola:

```
current_url = q.get()
```