Trabajo de Teoría Asignado: Crawlers.

Proyecto de Laboratorio Propuesto: Elaboración de un rastreador web o crawler.

Breve descripción del proyecto: Hemos enfocado nuestro proyecto en la elaboración y programación de un rastreador web, con el fin de aplicarlo dentro de sitios web para que navegue en ellos y extraigo la información que necesita. Las páginas que hemos seleccionado para esta extracción han sido:

- La Wikipedia en inglés: https://en.wikipedia.org/wiki/Main_Page
- La página de la ESI: https://esi.uclm.es

Hay varias formas posibles de programarlo, en función del lenguaje que uses. Lo más habitual es realizar el web scraping con Python y es el lenguaje de programación que hemos seleccionado. Otra forma es mediante Octoparse.

CRAWLER EN PYTHON:

¿POR QUÉ SE NECESITA UN WEB CRAWLER?

Imagínese un universo en el que Google no exista. ¿Cuánto tiempo crees que te tomaría encontrar una receta vegetariana en la web? Cada día se generan en línea cantidades astronómicas de datos, en concreto 2,5 quintillones de bytes. Sin la ayuda de motores de búsqueda como Google, la tarea de encontrar lo que buscas en Internet se convertiría en una verdadera locura, como intentar encontrar una aguja en un pajar. Los motores de búsqueda son herramientas fundamentales que nos permiten indexar y buscar, en la web, millones de páginas en un instante.

Cómo se construye un web crawler con codificación:

Algunos pasos generales para realizar un crawlers son los siguientes, los cuales nos han ayudado a codificar el nuestro:

1) Importa las bibliotecas necesarias: Necesitarás importar la biblioteca "requests" y "beautifulsoup4" para realizar solicitudes HTTP y analizar el HTML obtenido.
2) Obtén la página web que deseas rastrear: Utiliza la biblioteca "requests" para enviar una solicitud HTTP y obtener el contenido HTML de la página.
3) Analiza el contenido HTML de la página: Utiliza la biblioteca "beautifulsoup4" para analizar el HTML y extraer la información que necesitas.
4) Encuentra los enlaces de la página: Utiliza la función "find_all" de BeautifulSoup para encontrar todos los enlaces de la página y guardarlos en una lista.
5) Visita los enlaces y repite el proceso: Itera a través de los enlaces encontrados y repite los pasos 2 a 4 para cada uno de ellos.
6) Almacena la información recolectada: Puedes almacenar la información recolectada en una base de datos o en un archivo CSV.

Con respecto a nuestro crawler hemos implementado los siguientes módulos:

```
import requests, colorama
from bs4 import BeautifulSoup
from colorama import Fore, Style
import sqlite3
```

- “Requests” como bien se ha mencionado antes es un módulo que permite realizar solicitudes HTTP en Python y se utilizar para acceder al contenido de sitios web.
- “colorama” nos sirve para imprimir en la consola más esteticamente y legible, además de añadir colores y estilos al texto mostrado por pantalla de la Wikipedia en inglés.
- “BeatifulSoup” se utiliza para el análisis de documentos HTML y XML.
- “sqlite3” nos permite trabajar con bases de datos SQLite ya que después de analizar ambas páginas se almacenan en una base de datos.

A la hora de realizar la extracción de las páginas web, lo hemos hecho de forma distinta de una a otra:

EXTRACCIÓN DE INFORMACIÓN

Dentro de una función denominada: 'obtencionPaginaWebWiki' se obtiene el contenido HTML de la página relacionado con la url insertada en la variable denominada "url". Después se usa el módulo requests para una solicitud GET a la URL y así se puede obtener el contenido de la página web. El resultado de esta solicitud se almacena en la variable "response".
A continuación, se crea el objeto BeautifulSoup y se configura para parsear HTML. Y finalmente se devuelve el objeto soup para que el contenido se encuentre listo para ser procesado por el resto del programa.

```
def obtencionPaginaWebWiki():

    url = 'https://en.wikipedia.org/wiki/Main_Page'
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    return soup;
```

Para la página de la ESI, se hace lo mismo, pero cambiando la url por la de la ESI:

```python
def obtencionPaginaWebESI():

    # Hacemos la petición HTTP a la página web de la ESI
    url = 'https://esi.uclm.es'
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")

    return soup
```

Con respecto a la extracción de la información el proceso es el siguiente:

Este código define una función llamada “getInfoAndStorage” que toma como entrada dos objetos BeautifulSoup: soup y soup2. Estos objetos se crean a partir del contenido HTML de dos páginas web diferentes. Esta función busca y recupera información específica de la página de inicio de Wikipedia en inglés. Se usa la función find() de BeautifulSoup para encontrar el elemento HTML real en la página y luego text.strip() para extraer el texto de ese elemento y almacenarlo en una variable.
A continuación, se explican las diferentes secciones de la página que la función busca y almacena en variables:

- Título principal de la página
- Sección de bienvenida de la página
- Introducción de la página
- Contenido de las noticias
- Contenido del apartado "Did you know..."
- Contenido del apartado "Today's featured picture"
- Contenido del apartado "On this day"
- Contenido de la sección "Other areas of Wikipedia"
- Contenido de la sección "Wikipedia's sister projects"
- Contenido de la sección "Wikipedia languages"
- Además, la función también busca y almacena la URL de varias imágenes específicas de la página, incluyendo la imagen destacada del día.

```python
def obtencionInfoYAlmacenamiento(soup,soup2):

    titulo = soup.find('title').text

    bienvenida_section = soup.find('div', {'id': 'mp-welcomecount'})
    bienvenida_content = bienvenida_section.text.strip()

    introduccion_element = soup.find('div', {'id': 'mw-content-text'}).find('p').text
    if introduccion_element:
        introduccion = introduccion_element
    else:
        introduccion = None

    #Buscamos el contenido de las noticias
    noticias_section = soup.find('div', {'id': 'mp-itn'})
    noticias = noticias_section.text.strip()

    # Buscamos el contenido del apartado "did you know..."" y obtener su contenido
    did_you_know_section = soup.find('div', {'id': 'mp-dyk'})
    did_you_know_content = did_you_know_section.text.strip()
```

```python
# Buscamos el contenido del apartado "Today's featured picture" y obtener su contenido
featured_picture_section = soup.find('div', {'id': 'mp-tfp'})
featured_picture_content = featured_picture_section.text.strip()

# Buscamos el contenido del apartado "On this day" y obtener su contenido
on_this_day_section = soup.find('div', {'id': 'mp-otd'})
on_this_day_content = on_this_day_section.text.strip()

# Encontrar la sección "Other areas of Wikipedia" y obtener su contenido
other_areas_section = soup.find('div', {'id': 'mp-other-content'})
other_areas_content = other_areas_section.text.strip()

# Encontrar la sección "Wikipedia's sister projects" y obtener su contenido
wiki_projects_section = soup.find('div', {'id': 'mp-sister-content'})
wiki_projects_content = wiki_projects_section.text.strip()

# Encontrar la sección "Wikipedia languages" y obtener su contenido
wiki_languages_section = soup.find('div', {'class': 'wikipedia-languages nourlexpansion'})
wiki_languages_content = wiki_languages_section.text.strip()
```

```python
links = soup.find_all('img')
contador = 0
for imagen in links:
    img = imagen.get('src')
    contador += 1
    # print(img)    para imprimir todas las url de las imagenes de la pawina web.
    if contador == 8:
        imagen_destacada = img;
    if contador == 5:
        imagen_dyk = img;
    if contador == 7:
        imagen_otd = img;
    if contador == 4:
        imagen_ftfa = img;
    if contador == 6:
        imagen_itn = img;
```

La extracción de la información de la página de la ESI es algo distinta:
La sección que tiene una clase mkd-wrapper, utilizando la función find() de BeautifulSoup para buscar el elemento HTML con esa clase y luego la función text.strip() para extraer su contenido de texto. Esta sección selecciona todo el contenido y es más rápido que ir parte por parte como la anterior.

```
titulo2 = soup2.find('title').text

esi_section = soup2.find('div', {'class': 'mkd-wrapper'})
esi_content = esi_section.text.strip()
```

Finalmente se devuelve un return con todas las variables anteriormente mencionadas para posteriores funciones que son para imprimir la información y almacenarla en la base de datos.

IMPRESIÓN DE LA INFORMACIÓN

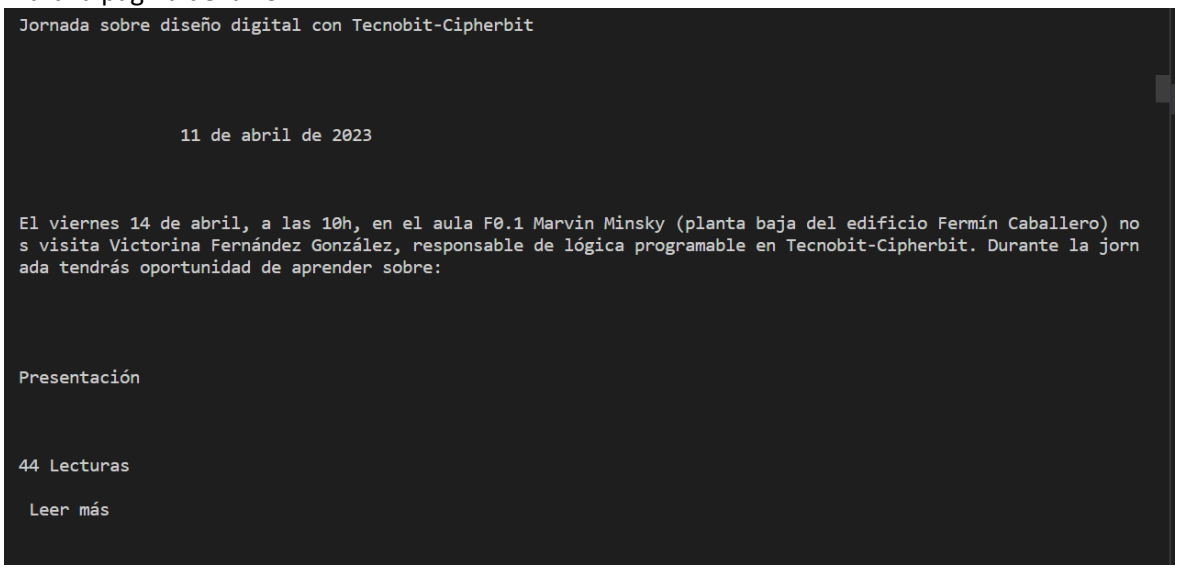
Para imprimir la información se ha implementado una función denominada “imprimirContenido” que tiene como parámetros de entradas las variables anteriormente mencionadas con la información y mediante printfs se muestra todo:

```
def imprimirContenido(titulo, bienvenida_content,introduccion,noticias,did_you_know_content,featured_pictu

    # Imprimimos los resultados
    print(Fore.MAGENTA+'Título:\n'+Style.RESET_ALL, titulo)
    print(Fore.MAGENTA+'\nBienvenida:\n'+Style.RESET_ALL, bienvenida_content)
    print(Fore.MAGENTA+"\nFrom today's featured article:"+Style.RESET_ALL+"\nURL imagen: "+Fore.GREEN+imag
    print(Fore.MAGENTA+'\nIn the news:'+Style.RESET_ALL+'\nURL imagen: '+Fore.GREEN+imagen_itn+Style.RESET

    print(Fore.MAGENTA+"\nDid you know ...:"+Style.RESET_ALL+"\nURL imagen: "+Fore.GREEN+imagen_dyk+Style.
    print(Fore.MAGENTA+"\nOn this day:"+Style.RESET_ALL+"\nURL imagen: "+Fore.GREEN+imagen_otd+Style.RESET

    print(Fore.MAGENTA+"\nToday's featured picture:"+Style.RESET_ALL)
    if imagen_destacada is not None:
        print('URL: '+Fore.GREEN+imagen_destacada+Style.RESET_ALL)
        print('\n'+featured_picture_content)
    else:
        print('No se encontró ninguna imagen destacada')

    print(Fore.MAGENTA+"\nOther areas of Wikipedia:\n"+Style.RESET_ALL, other_areas_content)

    print(Fore.MAGENTA+"\nWikipedia's sister projects:\n"+Style.RESET_ALL, wiki_projects_content)

    print(Fore.MAGENTA+"\nWikipedia languages:\n"+Style.RESET_ALL, wiki_languages_content)

    print(Fore.MAGENTA+"\n\nCONTENIDO PAGINA WEB ESI:\n-------------------------------\n"+Style.RESET_ALL

    colorama.deinit()
```

La información al ejecutar se muestra de la siguiente forma:
-Para la página de la Wikipedia en inglés

Y así con el resto de la información.

-Para la página de la ESI:



Y así con el resto de la información.

ALMACENAMIENTO EN BASE DE DATOS:

El código es una función que toma varios argumentos (variables) y los inserta en una tabla de una base de datos SQLite. La función primero establece una conexión a la base de datos y crea un cursor para ejecutar operaciones en la base de datos. A continuación, la función inserta los valores de las variables en la tabla "datos" utilizando la sentencia SQL "INSERT INTO". Luego, la función realiza una consulta SELECT para recuperar los datos de la tabla y los imprime en la consola. Después de eso, la función elimina todos los datos de la tabla utilizando la sentencia SQL "DELETE FROM", guarda los cambios y cierra la conexión con la base de datos. En resumen, la función se utiliza para insertar y recuperar datos de una base de datos SQLite.

```python
def meterDatosBBDD(titulo, bienvenida_content,introduccion,noticias,did_you_know_content,featured_picture_
    # Crear una conexión a la base de datos
    conn = sqlite3.connect('BBDDcrawler.db')

    # Crear un cursor para la base de datos
    c = conn.cursor()

    """
    # Crear una tabla para almacenar los datos
    c.execute('''CREATE TABLE datos (titulo text, bienvenida_content text,introduccion,noticias text,did_y
    """


    # Insertar datos en la tabla
    c.execute("INSERT INTO datos VALUES (?, ?, ?,?,?,?,?,?,?,?,?,?,?,?,?)", (titulo, bienvenida_content,

    # Guardar los cambios en la base de datos
    conn.commit()

    # Ejecutar la consulta
    c.execute('SELECT titulo, bienvenida_content,introduccion,noticias,did_you_know_content,featured_pictu

    # Recuperar los datos
    datos = c.fetchall()
    for fila in datos:
        print(fila)

    c.execute("DELETE FROM datos")
    conn.commit()
```

Para comprobar si el almacenamiento es el preciso lo mostramos y este es el resultado:

```
('Wikipedia, the free encyclopedia', 'Welcome to Wikipedia,\nthe free encyclopedia that anyone can edit.\n6,645
,455 articles in English', "Logan is an 1822 Gothic novel by American writer John Neal (depicted). The book is
inspired by the true story of Mingo leader Logan, but weaves a fictionalized story set just before the Revoluti
onary War. It depicts the genocide of Native Americans as the heart of the American story and follows a long ca
st of characters connected to each other in a complex web of overlapping love interests, family relations, rape
, and (sometimes incestuous) sexual activity. Scholars criticize the story's profound excessiveness and incoher
ence, but praise its pioneering and successful experimentation with psychological horror, verisimilitude, sexua
l guilt in male characters, impacts of intergenerational violence, documentation of interracial relationships,
and intersections between sex and violence on the American frontier. The novel is considered important by schol
ars studying the roles of Gothic literature and Indigenous identities in fashioning an American national identi
ty. (Full\xa0article...)\n", "JUICE's trajectory\n\nIn Sudan, at least 185 people die in clashes between rival
factions of the military government.\nR21/Matrix-M, a proven-effective malaria vaccine, is approved for use in
Ghana.\nThe European Space Agency launches the Jupiter Icy Moons Explorer (JUICE) to study Ganymede, Europa and
  Callisto (trajectory pictured).\nIn the Myanmar civil war, the military junta's air force kills  at least 130
 civilians in Pazigyi.\n\nOngoing: \nFrench pension reform unrest\nIsraeli judicial reform protests\nRussian in
vasion of Ukraine\nRecent deaths: \nRabey Hasani Nadwi\nFreddie Scappaticci\nKeshub Mahindra\nKarl Berger\nCrai
g Breen\nElisabeth Kopp\n\nNominate an article", 'Letitia Tyler\n\n... that Letitia Christian Tyler (pictured)
was the first United States first lady to die in the role?\n... that Nationalist China\'s own Northeastern Army
 captured Chiang Kai-shek to convince him to end the civil war against the Chinese Communist Party?\n... that t
he 2018 book The Longevity Diet claims that a "fast-mimicking diet" increases lifespan and healthspan?\n... tha
t Rihanna and Dua Lipa participated in #BlueforSudan to bring attention to the 3\xa0June 2019 Khartoum massacre
?\n... that the periodic comet 323P/SOHO approaches the Sun at a distance of 0.04\xa0AU, nearer than any other
numbered comet, every 4.15 years?\n... that the Berlin embassy of the Italian Social Republic published the wee
kly La Voce della Patria from 1943 to 1944 for distribution among Italian Military Internees in Germany?\n... t
hat Francois Massaquoi, who studied economics at New York University, later led the Lofa Defense Force during t
he First Liberian Civil War?\n... that The Noble Fisherman unusually places\xa0Robin Hood\xa0in the seaside tow
n of\xa0Scarborough, and he ends up fighting French pirates?\n\nArchive\nStart a new article\nNominate an artic
```

MAIN

Por último, en el Main para que todo funcione de forma correcta se realiza lo siguiente:

```python
def main():
    soup = obtencionPaginaWebWiki()
    soup2 = obtencionPaginaWebESI()
    (titulo, bienvenida_content,introduccion,noticias,did_you_know_content,featured_picture_content,on_thi
    imprimirContenido(titulo, bienvenida_content,introduccion,noticias,did_you_know_content,featured_pictu
    meterDatosBBDD(titulo, bienvenida_content,introduccion,noticias,did_you_know_content,featured_picture_
```

BIBLIOGRAFÍA:

https://www.octoparse.es/blog/guía-paso-a-paso-para-construir-un-web-crawler-para-principiantes

https://www.scrapingbee.com/blog/crawling-python/

https://secnot.com/web-crawler-con-scrapy.html