Actividad Extra 12

Qué es web scraping?

El web scraping es un proceso que consiste en extraer información de sitios web de forma automática utilizando herramientas y técnicas especializadas. Este proceso permite obtener datos estructurados o semiestructurados directamente de las páginas web para ser utilizados en diversos contextos, como análisis de datos, investigación, desarrollo de aplicaciones o incluso automatización de tareas repetitivas.

En términos técnicos, el web scraping implica acceder al código HTML de un sitio web, identificar los elementos relevantes (como texto, imágenes o enlaces) y extraer esa información para almacenarla o procesarla según las necesidades del usuario. Las herramientas más comunes para esta tarea incluyen bibliotecas de programación, como Beautiful Soup o Scrapy en Python, que facilitan la manipulación de contenido web.

Es importante destacar que, aunque el web scraping tiene múltiples aplicaciones útiles, debe realizarse respetando las políticas de uso de los sitios web y las regulaciones legales, para evitar conflictos relacionados con el uso no autorizado de los datos.

Realizar una prueba en python para dos librerías diferentes

a) Beautiful Soup

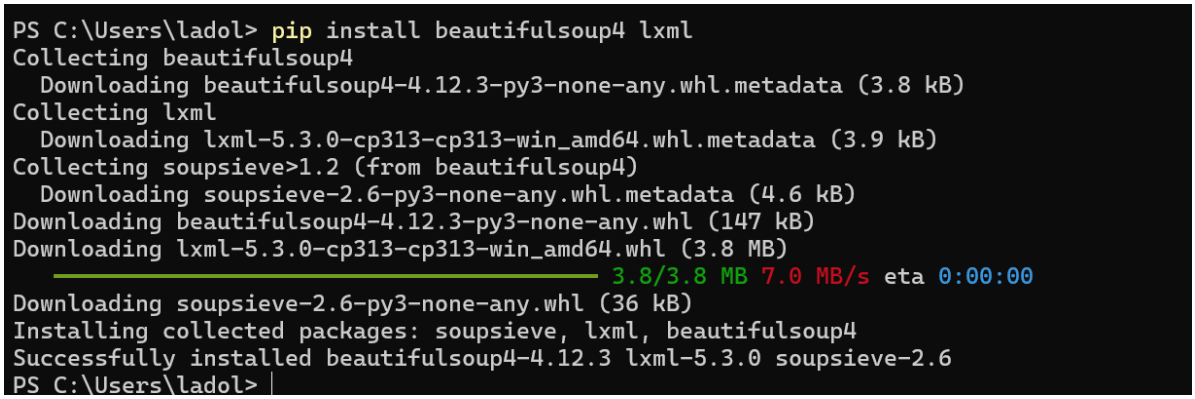

Figura 1: Instalación de las librería

Código de prueba

```python
# Generado con ayuda de Chatgpt

from bs4 import BeautifulSoup
import requests

url = "https://www.epn.edu.ec/"

# Realizar la solicitud GET
response = requests.get(url)

# Verificar que la solicitud fue exitosa
if response.status_code == 200:
    # Parsear el contenido HTML
    soup = BeautifulSoup(response.content, "lxml")

    # Extraer elementos, por ejemplo, títulos (h1, h2, etc.)
    titles = soup.find_all("h1")

    for i, title in enumerate(titles, 1):
        print(f"Título {i}: {title.text.strip()}")
else:
    print(f"Error al acceder a la página: {response.status_code}")
```

```
Título 1:
Título 2: SÍGUENOS
Título 3: LLÁMANOS
```

Título 4: UBICACIÓN

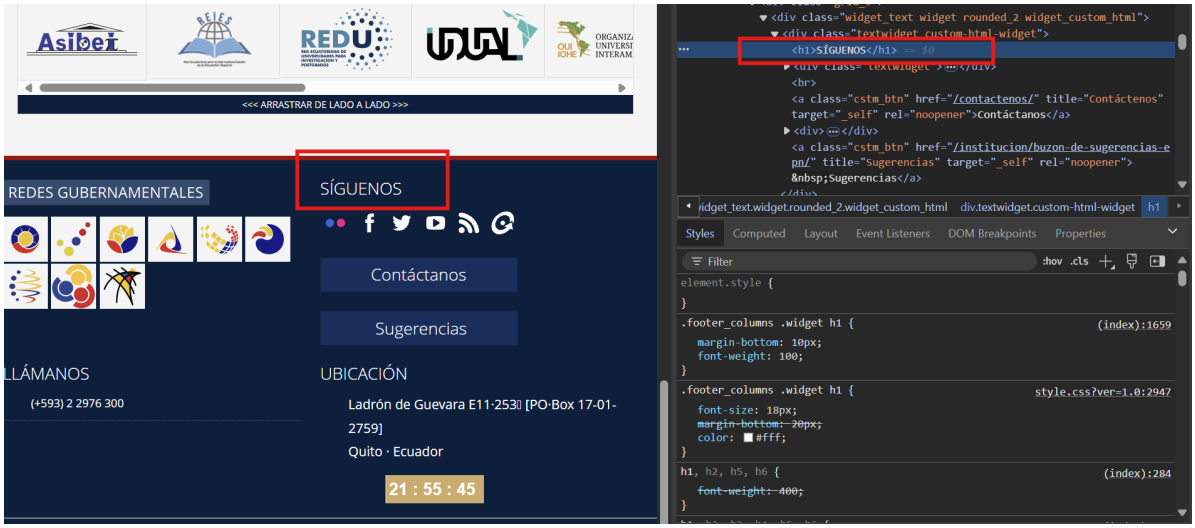

Figura 2: Evidencia en la página web

b) Scrapy

```
PS C:\Users\ladol> pip install scrapy
Collecting scrapy
  Downloading Scrapy-2.12.0-py2.py3-none-any.whl.metadata (5.3 kB)
Collecting Twisted>=21.7.0 (from scrapy)
  Downloading twisted-24.11.0-py3-none-any.whl.metadata (20 kB)
Collecting cryptography>=37.0.0 (from scrapy)
  Downloading cryptography-44.0.0-cp39-abi3-win_amd64.whl.metadata (5.7 kB)
Collecting cssselect>=0.9.1 (from scrapy)
  Downloading cssselect-1.2.0-py2.py3-none-any.whl.metadata (2.2 kB)
Collecting itemloaders>=1.0.1 (from scrapy)
  Downloading itemloaders-1.3.2-py3-none-any.whl.metadata (3.9 kB)
Collecting parsel>=1.5.0 (from scrapy)
  Downloading parsel-1.9.1-py2.py3-none-any.whl.metadata (11 kB)
Collecting pyOpenSSL>=22.0.0 (from scrapy)
  Downloading pyOpenSSL-24.3.0-py3-none-any.whl.metadata (15 kB)
Collecting queuelib>=1.4.2 (from scrapy)
  Downloading queuelib-1.7.0-py2.py3-none-any.whl.metadata (5.7 kB)
Collecting service-identity>=18.1.0 (from scrapy)
  Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB)
Collecting w3lib>=1.17.0 (from scrapy)
```

Figura 3: Instalación de las librería

Código que se uso en la prueba

```python
# Generado con ayuda de Chatgpt
import scrapy

class EjemploSpider(scrapy.Spider):
    name = "ejemplo"
    start_urls = [
        'https://quotes.toscrape.com', # sitio web diseñado para pruebas de scraping
    ]

    def parse(self, response):
        # Extraer títulos (por ejemplo, h1)
        for title in response.css('h1'):
            yield {
                'Título': title.get().strip(),
            }
```

```
PS C:\Users\ladol> scrapy startproject prueba_scrapy
New Scrapy project 'prueba_scrapy', using template directory 'C:\Users\l
s\scrapy\templates\project', created in:
    C:\Users\ladol\prueba_scrapy

You can start your first spider with:
    cd prueba_scrapy
    scrapy genspider example example.com
PS C:\Users\ladol> cd prueba_scrapy
PS C:\Users\ladol\prueba_scrapy> scrapy genspider ejemplo example.com
Created spider 'ejemplo' using template 'basic' in module:
  prueba_scrapy.spiders.ejemplo
PS C:\Users\ladol\prueba_scrapy> scrapy crawl ejemplo
```

Figura 4: Comandos utilizados en la prueba

Resultado de Scrapy

```
2025-01-04 22:15:13 [scrapy.core.scraper] DEBUG: Scraped from <200 https://quotes.toscrape.com/>
{'Título': '<h1>\n                <a href="/" style="text-decoration: none">Quotes to Scrape</a>\n            </h1>'}
2025-01-04 22:15:13 [scrapy.core.engine] INFO: Closing spider (finished)
2025-01-04 22:15:13 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
```

Figura 5: Resultado

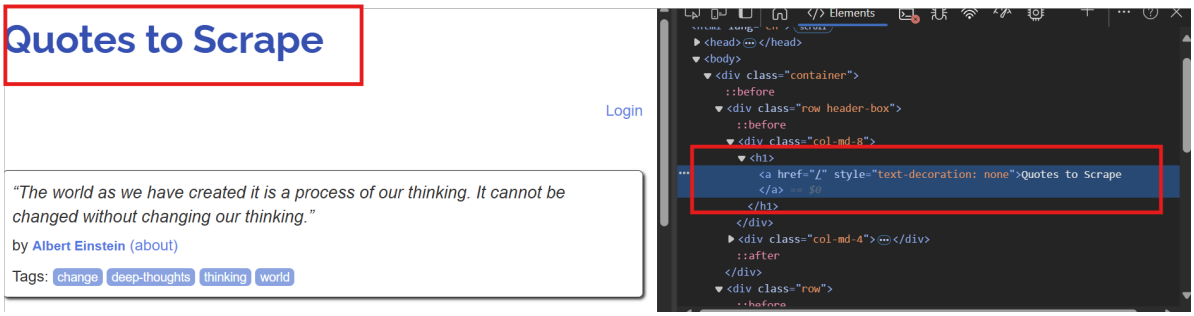

Figura 6: IEvidencia del resultado de scrapy