Logo

Description automatically generatedLogo

Description automatically generated with medium confidence

*Practica: 05*

*Nombre de la Practica: Request y Beautiful soap*

Instituto Politécnico Nacional.

Escuela Superior de Cómputo.

Licenciatura en ciencia de datos.

Nombre de la materia: Desarrollo de Aplicaciones para el Análisis de Datos

Grupo: 4AV1

Profesora: Sandra Luz Morales Guitron

Lopez Mendez Emiliano

INDICE

[INTRODUCCION 3](#_Toc182592679)

[DESAROLLO 3](#_Toc182592680)

[Protocolo HTTP y librería REQUESTS 4](#_Toc182592681)

[Codigo Ejemplo 7](#_Toc182592682)

[Web scraping con REQUESTS Y BEAUTIFULSOUP 8](#_Toc182592683)

[CONCLUCIONES 17](#_Toc182592684)

# INTRODUCCION

En esta práctica, nos sumergimos en el mundo del web scraping utilizando Python y dos herramientas clave: Requests y BeautifulSoup. El objetivo fue extraer información estructurada de un sitio web, aprendiendo a navegar por el árbol DOM y manipular datos en formato HTML. Este enfoque es esencial para tareas como análisis de datos, recopilación automatizada de información y creación de aplicaciones basadas en datos dinámicos de la web. A lo largo de la práctica, se exploraron conceptos fundamentales de HTTP, incluyendo solicitudes GET y manejo de encabezados, lo que permitió sortear restricciones comunes en los sitios web modernos.

# DESAROLLO

**Herramientas principales:**

1. **Requests**: Se utilizó para realizar solicitudes HTTP, recuperando el contenido HTML de una página web. Esto involucró configurar encabezados como el User-Agent para simular el comportamiento de un navegador y evitar bloqueos.
2. **BeautifulSoup**: Ayudó a transformar el contenido HTML en un árbol DOM navegable, facilitando la búsqueda y extracción de elementos específicos utilizando métodos como find y find\_all.

**Pasos principales del análisis:**

1. **Descarga de datos**:
   * Se realizó una solicitud GET a una página específica de MercadoLibre, simulando un navegador para evitar restricciones.
   * El contenido HTML se almacenó localmente para inspección y reutilización futura.
2. **Extracción de elementos clave**:
   * Se obtuvo el título de la página y el nombre del producto utilizando etiquetas <title> y <h1> con clases específicas.
   * Se analizaron precios actuales y anteriores, limpiándolos y convirtiéndolos a formato numérico para posibles análisis posteriores.
   * Se extrajeron características del producto, como capacidad de almacenamiento, tipo de consola y detalles adicionales desde <div> con clases relevantes.
3. **Análisis de calificaciones y opiniones**:
   * Se identificó la calificación promedio de los usuarios (por ejemplo, 5 estrellas) y el número de opiniones.
   * Este análisis es útil para determinar la popularidad y confiabilidad del producto.
4. **Imágenes del producto**:
   * Se localizaron elementos <img> que contenían enlaces a las imágenes del producto. Se extrajeron atributos como src, que contienen los enlaces directos a las imágenes.

**Desafíos encontrados:**

* La estructura HTML de la página incluía múltiples niveles de etiquetas y clases anidadas, lo que requirió identificar las clases específicas para cada elemento deseado.
* La página actualizó dinámicamente ciertos datos, como calificaciones, lo que exigió revalidar los selectores CSS utilizados.

## Protocolo HTTP y librería REQUESTS

A continuación se lleva acabo un ejemplo de los entresijos del protocolo HTTP, lo cual es fundamental para sortear los obstáculos que nos pondrán las webs que queramos scrapear en el futuro. Además, veremos cómo acceder a los datos, tanto de la petición del cliente como de la respuesta del servidor, con la librería requests.

A black screen with white text

Description automatically generated

Se importa la librería requests, que se utiliza para hacer solicitudes HTTP.

A screenshot of a computer program

Description automatically generatedDefine la URL de un producto de Amazon y un encabezado User-Agent para imitar una solicitud desde un navegador común, lo cual puede ayudar a evitar bloqueos de Amazon por considerar la solicitud como automatizada.

A screen shot of a computer screen

Description automatically generated

Envía una solicitud GET a la URL de Amazon con los encabezados especificados. La variable res.status\_code guarda el código de estado de la respuesta, y un valor de 200 indica que la solicitud fue exitosa.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

res.reason devuelve una descripción del estado de respuesta (por ejemplo, 'OK'). res.ok es un valor booleano que indica si la solicitud fue exitosa (True si el código de estado es 200).

A screen shot of a computer

Description automatically generated

Abre un archivo llamado codigo\_600.html y guarda el contenido HTML de la página obtenida. Esto permite ver el código HTML del producto en Amazon sin necesidad de una conexión en tiempo real.

A black and grey line

Description automatically generated Aquí podemos ver como ya se guardo el código en nuestra maquina

A screenshot of a computer program

Description automatically generated

El contenido de la pagina html

A computer screen shot of a backpack

Description automatically generated

A screenshot of a computer

Description automatically generated

Podemos ver como ya solo es el contenido html guardado en nuestra computadora y no es la pagina web

A screen shot of a computer

Description automatically generated

Muestra las cookies establecidas por Amazon en la respuesta. Estas cookies pueden ser necesarias para hacer solicitudes posteriores a Amazon.

A screen shot of a computer

Description automatically generated

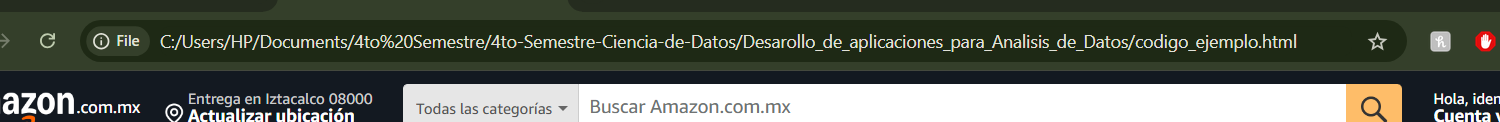
Muestra los encabezados de la solicitud que se envió, incluidos el User-Agent, la codificación aceptada, y la conexión.

### Codigo Ejemplo

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

## Web scraping con REQUESTS Y BEAUTIFULSOUP

Sitio web usado: <https://www.mercadolibre.com.mx/consola-switch-oled-mario-kart-8-3-meses-nintendo-online/p/MLM37283984?pdp_filters=deal:MLM1169088-1#wid=MLM3453172814&sid=search&searchVariation=MLM37283984&position=3&search_layout=grid&type=product&tracking_id=3d0340a0-4b9c-4a6b-9084-e10f32fe199d&DEAL_ID=flag%20gaming%20lo%20mas%20vendido&S=landingHubconsolas-y-videojuegos&V=11&T=CarouselDynamic-home&L=LO-MAS-VENDIDO-EN-CONSOLAS&deal_print_id=936f94d0-a371-11ef-b341-15351d57c41d&c_id=carouseldynamic-home&c_element_order=undefined&c_campaign=LO-MAS-VENDIDO-EN-CONSOLAS&c_uid=936f94d0-a371-11ef-b341-15351d57c41d>

A black screen with colorful text

Description automatically generated

**import requests:**

* La librería requests permite realizar solicitudes HTTP para obtener datos de una página web. Es fundamental para descargar el contenido HTML de un sitio.

**from bs4 import BeautifulSoup:**

* BeautifulSoup es una librería de Python que facilita la extracción y análisis de datos del HTML obtenido.
* Proporciona herramientas para navegar y buscar dentro del árbol DOM del documento HTML.

A black background with orange and white text

Description automatically generated

Se establece la URL de un producto específico de MercadoLibre y se define un user-agent que simula ser un navegador web.

A screen shot of a computer code

Description automatically generated

Se realiza una solicitud GET a la URL, enviando el user-agent como parte de los headers.

A screenshot of a computer

Description automatically generated

BeautifulSoup convierte el contenido HTML obtenido (res.text) en un árbol DOM que puede ser explorado y analizado. Se imprime todo el contenido HTML de la página. Es muy extenso porque incluye toda la estructura y los elementos HTML de la página.

A screen shot of a computer

Description automatically generated

soup.title devuelve el elemento <title> del HTML.

soup.title.text extrae el texto dentro del elemento <title>.

Aquí imprimimos el titulo de la pagina web.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Muestra todas las funciones y atributos disponibles en el objeto BeautifulSoup. Imprime unna lista extensa de métodos y atributos, como find, find\_all, text, attrs, entre otros. Nosotros usaremos ‘find’ para encontrar los atributos que queremos al igual para encontrar las etiquetas con sus clases.

A screenshot of a computer

Description automatically generated

Busca el elemento <h1> con la clase CSS ui-pdp-title y extrae su texto. Cual aquí seria el nombre del producto vendido.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

Busca un <div> con la clase CSS ui-pdp-price\_\_second-line, que contiene el precio actual.Luego para extraer, limpiar y convertir en el precio actual:

Extrae el precio del elemento <span> y elimina comas y símbolos $.

Convierte el precio limpio a tipo float

A computer screen with text

Description automatically generated

A computer screen shot of a code

Description automatically generated

Hacemos lo mismo para extraer el precio anterior pero con etiquetas y clases diferentes.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Busca el <div> que contiene el ranking de usuarios (estrellas y número de opiniones) y devolvemos el contenido completo del <div> que muestra la calificación y opiniones de los usuarios. Luego extraemos la calificación promedio (por ejemplo, 5.0) de un elemento <span> con la clase CSS correspondiente. En el momento de hacer el código era de 5 estrellas pero actualizando la pagina web se movió a 4.5 estrellas

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Encuentra el elemento que contiene la imagen del producto. Con soup.find(class\_="ui-pdp-image ui-pdp-gallery\_\_figure\_\_image") se obtiene todo el contenido del elemento <img> que contiene la imagen principal. Al agregar ‘.attrs.’ al final extraemos todos los atributos de esa clase. Luego con ‘.get(“src”) extraemos el atributo src, que contiene el enlace a la imagen del producto.

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Extrae las características principales del producto desde un <div> con la clase CSS correspondiente y también extrae la descripción del producto desde un <div> con la clase CSS correspondiente.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

# CONCLUCIONES

Esta práctica demostró el poder del web scraping para la obtención de datos estructurados desde sitios web públicos. Requests y BeautifulSoup se consolidaron como herramientas esenciales, combinando simplicidad y flexibilidad para extraer información específica de documentos HTML. Aprendimos a lidiar con restricciones comunes de los servidores, como el bloqueo de solicitudes automatizadas, utilizando encabezados apropiados.

El uso de estas técnicas abre puertas a aplicaciones en análisis de mercado, monitoreo de productos, o incluso automatización de procesos. Sin embargo, también subrayó la importancia de realizar estas actividades respetando las políticas de uso de los sitios web y la ética en la recolección de datos. La experiencia adquirida fortalece nuestra capacidad para diseñar soluciones basadas en datos, listas para enfrentar problemas del mundo real en el ámbito del análisis y desarrollo de software.