



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
DESAROLLO	3
Protocolo HTTP y librería REQUESTS	4
Codigo Ejemplo.....	7
Web scraping con REQUESTS Y BEAUTIFULSOUP	8
CONCLUSIONES	17

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:

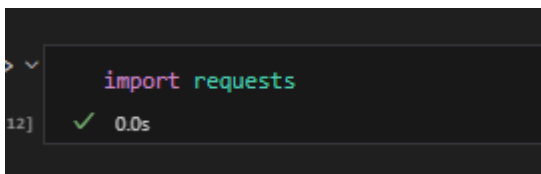
- Se localizaron elementos `` que contenían enlaces a las imágenes del producto. Se extrajeron atributos como `src`, que contienen los enlaces directos a las imágenes.

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.

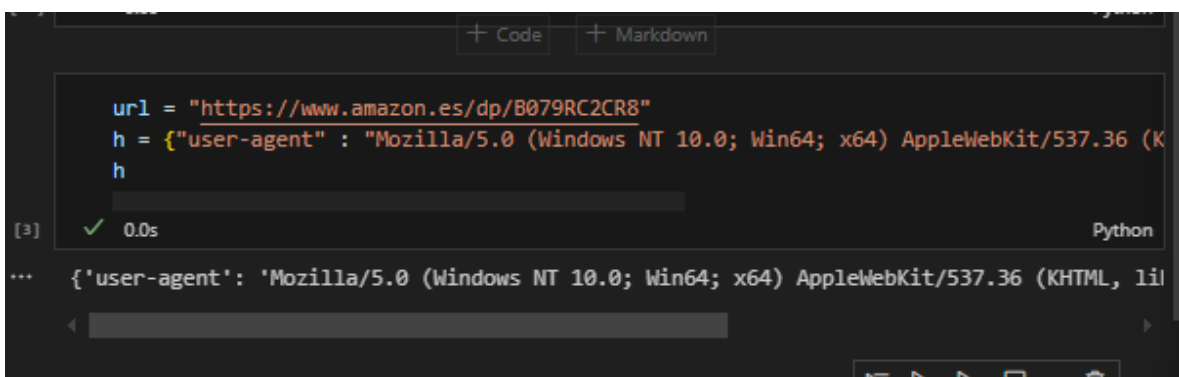
Protocolo HTTP y librería REQUESTS

A continuación se lleva a cabo 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`.



```
> import requests
[12] ✓ 0.0s
```

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



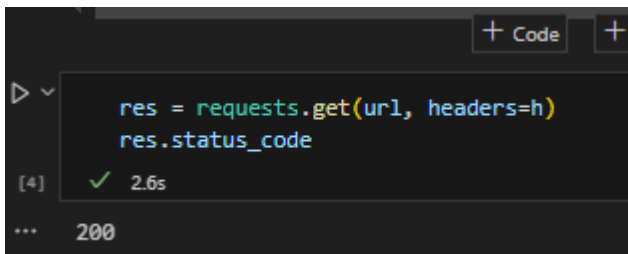
```
+ Code + Markdown

url = "https://www.amazon.es/dp/B079RC2CR8"
h = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.0.0 Safari/537.36"}

[3] ✓ 0.0s Python

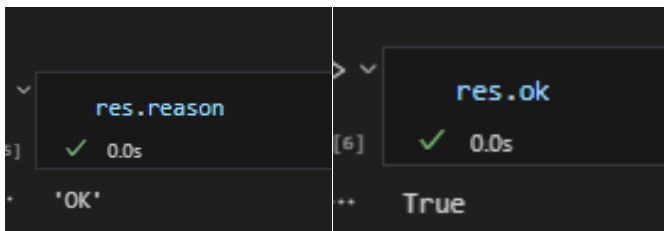
... {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.0.0 Safari/537.36'}
```

Define 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.



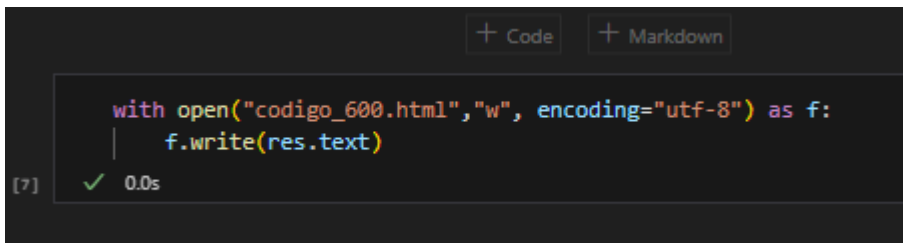
```
+ Code +  
res = requests.get(url, headers=h)  
res.status_code  
[4] ✓ 2.6s  
... 200
```

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.



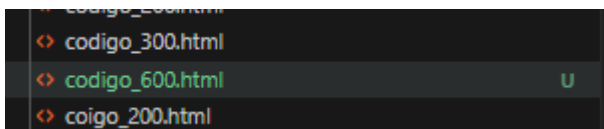
```
res.reason  
[5] ✓ 0.0s  
... 'OK'  
  
res.ok  
[6] ✓ 0.0s  
... True
```

`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).



```
+ Code + Markdown  
with open("codigo_600.html", "w", encoding="utf-8") as f:  
    f.write(res.text)  
[7] ✓ 0.0s
```

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.



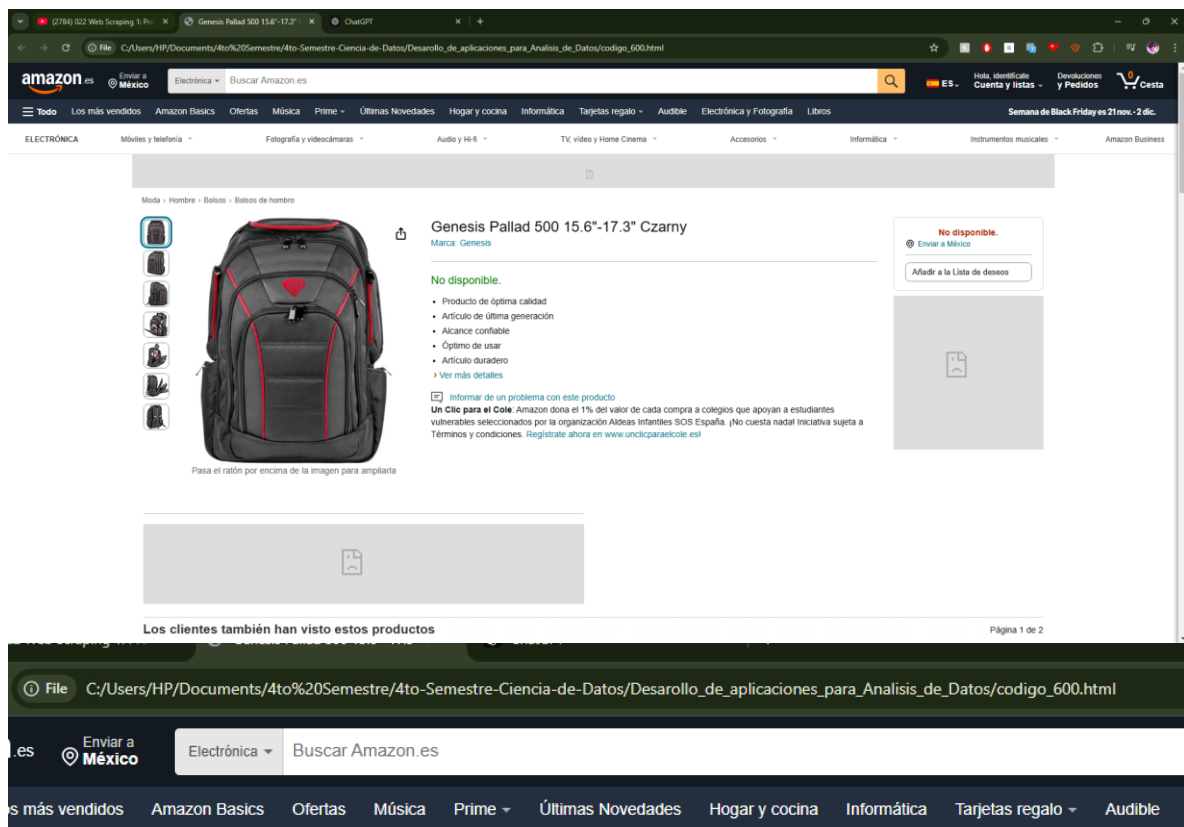
```
codigo_600.html  
codigo_300.html  
codigo_600.html U  
codigo_200.html
```

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

15 de noviembre de 2024

```
inal Help 4to-Semestre-Ciencia-de-Datos
web-scraping.ipynb M codigo_600.html U X
Desarrollo_de_aplicaciones_para_Analisis_de_Datos > codigo_600.html > ...
1 <!doctype html><html lang="es-es" class="a-no-js" data-19ax5a9jf="dingo"><!-- sp:feature:head-start -->
2 <head><script>var aPageStart = (new Date()).getTime();</script><meta charset="utf-8"/>
3 <!-- sp:end-feature:head-start -->
4 <!-- sp:feature:csm:head-open-part1 -->
5
6 <script type="text/javascript">var ue_t0=ue_t0||new Date();</script>
7 <!-- sp:end-feature:csm:head-open-part1 -->
8 <!-- sp:feature:cs-optimization -->
9 <meta http-equiv="x-dns-prefetch-control" content="on">
10 <link rel="dns-prefetch" href="https://images-eu.ssl-images-amazon.com">
11 <link rel="dns-prefetch" href="https://m.media-amazon.com">
12 <link rel="dns-prefetch" href="https://completion.amazon.com">
13 <!-- sp:end-feature:cs-optimization -->
14 <!-- sp:feature:csm:head-open-part2 -->
15 <script type="text/javascript">
16 window.ue_ihb = (window.ue_ihb || window.ueinit || 0) + 1;
17 if (window.ue_ihb === 1) {
18
19   var ue_csm = window,
20       ue_hob = +new Date();
21   (function(d){var e=d.ue.d.ue||{},f=Date.now||function(){return+new Date};e.d=function(b){return f().-(b?d.ue_t0)};e.stub=function(b,a){if(!b[a]){var c=[];b[a]
22
23     var ue_err_chan = 'jserr-rw';
24     (function(d,e){function h(f,b){if(!a.ec+a.mx&&f){a.ter.push(f);b=b||{};var c=f.logLevel||b.logLevel;c&&c!-k&&c!-m&&c!-n&&c!-p||a.ec++;c&&c!-k||a.ecf++;b
25     pec:0,ts:0,erl:[],ter:[],buffer:[],mxe:50,startTimer:function(){a.ts++;setInterval(function(){d.ue&&a.pec&a.ec&&d.uex("at");a.pec=a.ec,1E4)});l.skipTrace=1;h
26
27   var ue_id = '7YHY9GT688J5AwFHG87A',
28       ue_url = '/rd/uedata',
29       ue_navtiming = 1,
30       ue_mid = 'AIRKKUPIHCS9HS',
31       ue_sid = '260-9940912-0166430',
32       ue_sn = 'www.amazon.es',
33       ue_furl = 'fls-eu.amazon.es',
34       ue_surl = 'https://unagi-eu.amazon.com/1/events/com.amazon.csm.nexusclient.prod',
35       ue_int = 0;
```

El contenido de la pagina html



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

15 de noviembre de 2024

```
res.cookies
13] ✓ 0.0s Python
<RequestsCookieJar[Cookie(version=0, name='session-id', value='260-9940912-0166430', po
```

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

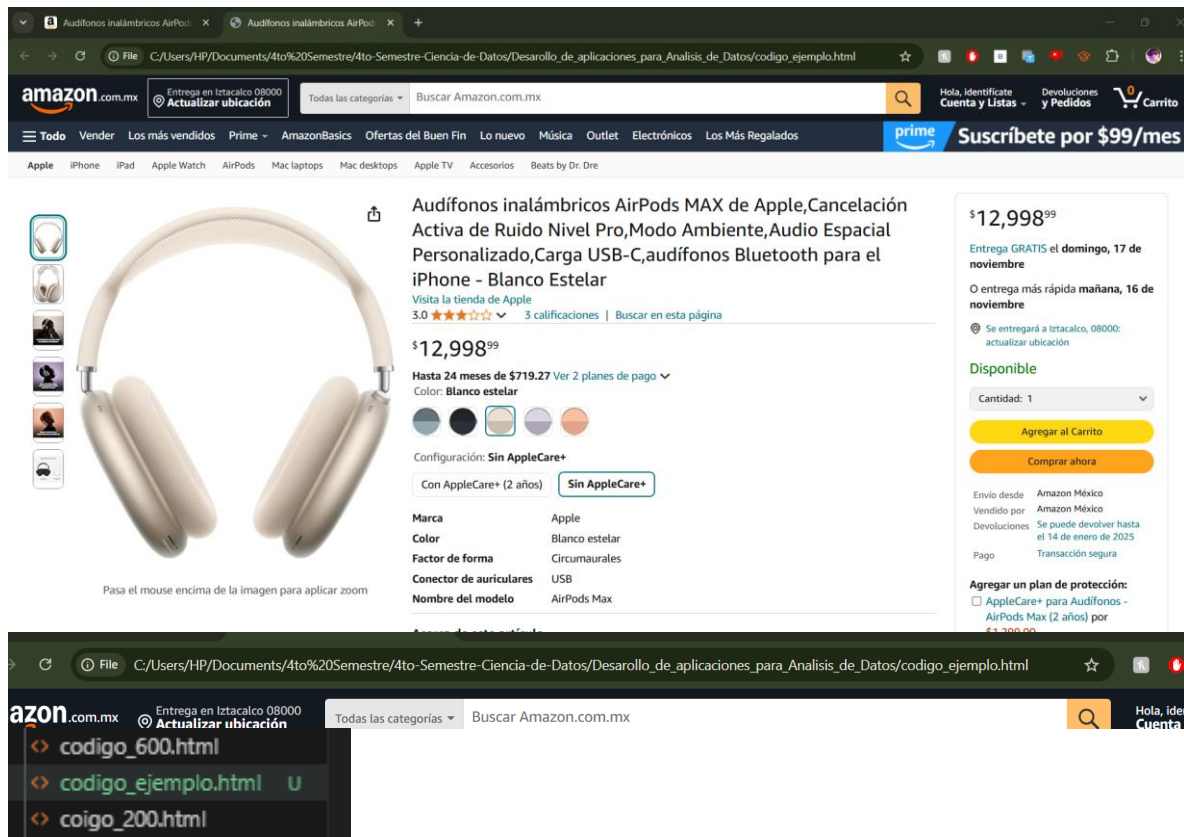
```
res.request.headers
✓ 0.0s Python
{'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, lil
```

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

Codigo Ejemplo

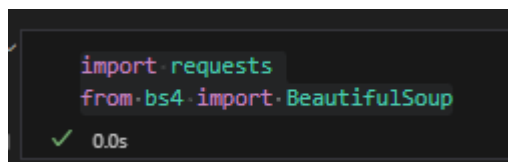
```
import requests
url_2 = "https://www.amazon.com.mx/gp/aw/d/B00GJBQ24F/?_encoding=UTF8&pd_rd_plhdr=t&aaxitk=9a34e33c2d6da1
h_2 = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
print(h_2)
res_2 = requests.get(url_2, headers=h_2)
print(res_2.status_code)
print(res_2.reason)
print(res_2.ok)
with open("codigo_ejemplo.html", "w", encoding="utf-8") as f:
    f.write(res_2.text)
print(res_2.cookies)
print(res_2.request.headers)
[3] ✓ 2.6s Python
... {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.
200
OK
True
<RequestsCookieJar[<Cookie session-id=137-9154715-5151438 for .amazon.com.mx/>, <Cookie session-id-time=2082
{'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.1
```

15 de noviembre de 2024



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

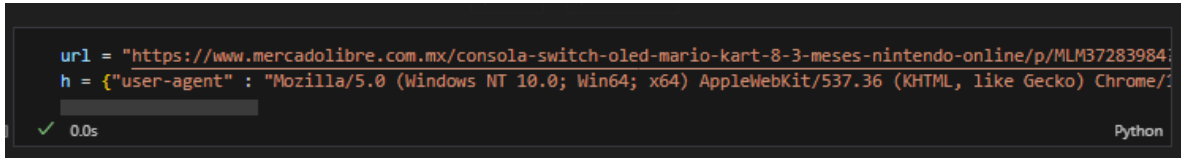


import requests:

- 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:

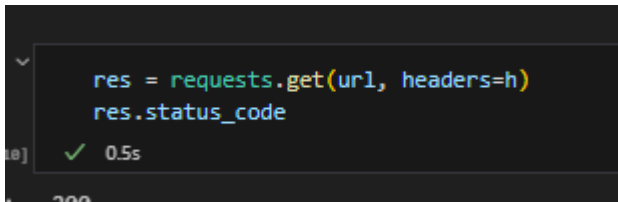
- BeautifulSoup es una librería de Python que facilita la extracción y análisis de datos del HTML obtenido.
- Proporciona herramientas para navegar y buscar dentro del árbol DOM del documento HTML.



```
url = "https://www.mercadolibre.com.mx/consola-switch-oled-mario-kart-8-3-meses-nintendo-online/p/MLM37283984?cid=MLM37283984"
h = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3930.162 Safari/537.36"}

✓ 0.0s Python
```

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



```
res = requests.get(url, headers=h)
res.status_code

✓ 0.5s
```

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

```

• soup = BeautifulSoup(res.text, "html.parser")
  print(soup)
✓ 0.1s Python

<!DOCTYPE html>

<html lang="es-MX">
<head><link href="https://www.google-analytics.com" rel="preconnect"/><link href="https://www.google.com" rel="pr
((()=>{var e,t,r={8122:(e,t,r)=>{"use strict";r.d(t,{a:()=>i});var n=r(944);function i(e,t){try{if(!e||"object"!=t
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="width=device-width, initial-scale=1.0, maximum-scale=5.0" name="viewport"/>
<meta content="True" name="HandheldFriendly"/>
<meta content="on" http-equiv="cleartype"/>
<meta content="samesite=true" name="browser-support"/>
<meta content="q0ocldHe-yAwZA9i2cpY3geKj8QDwqiwwkVk" name="csrf-token"/>
<link as="font" crossorigin="anonymous" data-head-react="true" href="https://http2.mlstatic.com/ui/webfonts/v3.0.
/*! * Web Fonts from Fontspring.com * * All OpenType features and all extended glyphs have been removed. * Fully
@font-face{font-family:'Proxima Nova';font-weight:300;font-display:swap;font-style:normal;src:url(https://http2.m
<style>@charset "utf-8";
/*
 * Navigation
 * @platform "mercadolibre"
 * @version 6.6.92
 * @author MercadoLibre.com
 */
@font-face{font-family:"navigation";src:url("https://http2.mlstatic.com/frontend-assets/ml-web-navigation/ui-navi
<style>a.nav-skip-to-main-content{border: 0;border-bottom-left-radius: 4px;border-bottom-right-radius: 4px;positi
...
    loadScripts(scripts);
  });
}
})(window, document);</script><script type="application/ld+json">{"name":"Consola Switch Oled + Mario Kart 8 + 3
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```

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.

```

• soup.title
  soup.title.text
✓ 0.0s

'Consola Switch Oled + Mario Kart 8 + 3 Meses Nintendo Online | Meses sin interés'

```

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

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

Aquí imprimimos el título de la página web.

```

<meta name="al:iphone:app_store_id" property="al:ios:app_store_id" content="463624852">
▼ <title> == $0
  "Consola Switch Oled + Mario Kart 8 + 3 Meses Nintendo Online | Meses sin interés"
</title>
<link rel="canonical" href="https://www.mercadolibre.com.mx/consola-switch-oled-mario-kar

```

```

dir(soup)

['ASCII_SPACES',
 'DEFAULT_BUILDER_FEATURES',
 'DEFAULT_INTERESTING_STRING_TYPES',
 'EMPTY_ELEMENT_EVENT',
 'END_ELEMENT_EVENT',
 'NO_PARSER_SPECIFIED_WARNING',
 'ROOT_TAG_NAME',
 'START_ELEMENT_EVENT',
 'STRING_ELEMENT_EVENT',
 '__bool__',
 '__call__',
 '__class__',
 '__contains__',
 '__copy__',
 '__deepcopy__',
 '__delattr__',
 '__delitem__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattr__',
 '__getattribute__',
 ...
 'stripped_strings',
 'tagStack',
 'text',
 'unwrap',
 'wrap']

```

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings](#)...

Muestra todas las funciones y atributos disponibles en el objeto BeautifulSoup. Imprime una 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.

```

soup.find("h1", class_="ui-pdp-title").text

'Consola Switch Oled + Mario Kart 8 + 3 Meses Nintendo Online'

```

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

```

<div class="ui-pdp-header__subtitle">...</div>
<div class="ui-pdp-header__title-container">flex
  <h1 class="ui-pdp-title">Consola Switch Oled + Mario Kart 8 + 3 Meses Nintendo Online</h1> ==
  <div class="ui-pdp-promotions-pill ui-pdp-highlights">...</div>
</div>

```

▼ Precio Normal

```

soup.find("div", class_="ui-pdp-price__second-line")

```

[131] ✓ 0.0s Python

```

<div class="ui-pdp-price__second-line"><span data-testid="price-part"><span aria-label="6430 pesos con 80 centavo

```

```

precio = soup.find("span", class_="andes-money-amount ui-pdp-price__part andes-money-amount--cents-superscript")
print(precio)
precio_float = float(precio)
print(precio_float)

```

[155] ✓ 0.0s Python

```

643080
643080.0

```

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 y elimina comas y símbolos \$.

Convierte el precio limpio a tipo float

```

6px" itemprop="offers" itemscope itemtype="http://schema.org/Offer"
role="img" aria-label="6419 pesos" aria-roledescription="Precio">
  flex
  <meta itemprop="price" content="6419">
  <span class="andes-money-amount__currency-symbol" itemprop="priceCurrency" aria-hidden="true">$</span>
  <span class="andes-money-amount__fraction" aria-hidden="true">
    6,419</span> == $0
  </span>
</span>
▶ <span class="ui-pdp-price second-line label ui-pdp-color--GREEN ui-p

```

▼ Precio anterior

```

precio_antes = soup.find("span", class_="andes-money-amount__fraction").text.replace(",","")
fprecio_antes = float(precio_antes)
print(fprecio_antes)

```

[14] ✓ 0.0s

```

8599.0

```

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

```

<img alt="Antes: 8599 pesos" data-bbox="135 85 859 115"/>
<span class="andes-money-amount__currency-symbol" aria-hidden="true">
  </span>
  <span class="andes-money-amount__fraction" aria-hidden="true">
    8,599
    </span>
  </div>

```

Ranking de usuarios

```

soup.find("div", class_="ui-pdp-header__info")
[161] ✓ 0.0s

<div class="ui-pdp-header__info"><a class="ui-pdp-review__label ui-pdp-re
  </a>
  <span class="ui-pdp-review__rating">
    5.0
  </span>
</div>

```

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 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

```

<div class="ui-pdp-header__info">
  <a href="/noindex/catalog/reviews/MLM37283983?noIndex=true&access=
    _all&modal=true&controlled=true" class="ui-pdp-review__label ui-pd
    view__label--link">
      <span aria-hidden="true" class="ui-pdp-review__rating">4.5
      </span>
      <span aria-hidden="true" class="ui-pdp-review__ratings">
        </span>
        <span class="andes-visually-hidden">Calificación 4.5 de 5. 8

```

```

    ✓ Imagen del producto
    + Code + Markdown

    soup.find(class_="ui-pdp-image ui-pdp-gallery__figure__image")
    [166] ✓ 0.0s Python
    ... _figure__image" data-index="0" data-zoom="https://http2.mlstatic.com/D_NQ_NP_614423-MLU76509896924_052024-F.webp"

    soup.find(class_="ui-pdp-image ui-pdp-gallery__figure__image").attrs
    [167] ✓ 0.0s Python
    ... {'data-zoom': 'https://http2.mlstatic.com/D_NQ_NP_614423-MLU76509896924_052024-F.webp',
    'data-index': '0',
    'width': '410',
    'height': '500',
    'fetchpriority': 'high',
    'decoding': 'sync',
    'src': 'https://http2.mlstatic.com/D_NQ_NP_614423-MLU76509896924_052024-O.webp',
    'srcset': 'https://http2.mlstatic.com/D_NQ_NP_2X_614423-MLU76509896924_052024-F.webp 2x',
    'class': ['ui-pdp-image', 'ui-pdp-gallery__figure__image'],
    'alt': 'Consola Switch Oled + Mario Kart 8 + 3 Meses Nintendo Online'}

    soup.find(class_="ui-pdp-image ui-pdp-gallery__figure__image").attrs.get("src")
    [168] ✓ 0.0s Python
    ... 'https://http2.mlstatic.com/D_NQ_NP_614423-MLU76509896924_052024-O.webp'
  
```

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 `` 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.

```

    i-pdp-gallery__label"> </label> flex
    <figure class="ui-pdp-gallery__figure"> flex
       == $0
    </figure>
  </span>
  
```

The screenshot shows a web browser with the title "Características del producto". Below the title, there is a table of specifications. A Python console at the bottom shows two queries using BeautifulSoup to extract data from the page.

```

soup.find("div", class_="ui-vpp-highlighted-specs__attribute-columns").text
[177] ✓ 0.0s Python
... 'Capacidad 64 GBTipo de consola HíbridaCantidad de controles incluidos 2Memoria RAM 64 GBCon Wi-Fi Sí'

soup.find("div", class_ = "ui-pdp-description pl-24 pr-24 ui-pdp-collapsible--is-collapsed").text
[179] ✓ 0.0s Python
... 'DescripciónCon tu consola HMG-S-KAASA tendrás entretenimiento asegurado todos los días. Su tecnología fue creada

```

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.

The screenshot shows a web browser with the title "Características del producto". Below the title, there is a table of specifications. The table has two columns: "Característica" and "Valor". The first row shows "Capacidad" with the value "64 GB".

```

<div class="ui-pdp-container__row ui-pdp-container__row--highlighted-specs-title">
  <h2 class="ui-vpp-text-alignment--left ui-pdp-color--BLACK ui-pdp-size--XLARGE ui-pdp-family--REGULAR highlighted-specs-title">Características del producto</h2>
</div>
<div class="ui-pdp-container__row ui-pdp-container__row--attributes">
  <div class="ui-vpp-highlighted-specs__attribute-columns">
    <div class="ui-vpp-highlighted-specs__attribute-columns__column">
      <div class="ui-pdp-container__row ui-vpp-highlighted-specs__attribute-columns__row">
        <div class="ui-vpp-highlighted-specs__key-value">
          <div aria-hidden="true" class="ui-pdp-background-color--GRAY_40 ui-vpp-highlighted-specs__filled-icon ui-vpp-highlighted-specs__filled-icon--ROUNDED">
            <div class="ui-vpp-highlighted-specs__key-value__labels">
              <p class="ui-pdp-family--REGULAR ui-vpp-highlighted-specs__key-value__labels_key-value"> == $0
                <span class="ui-pdp-color--BLACK ui-pdp-size--XSMALL ui-pdp-family--REGULAR">Capacidad</span>
                <span class="ui-pdp-color--BLACK ui-pdp-size--XSMALL ui-pdp-family--SEMIBOLD">64 GB</span>
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
<div class="ui-pdp-collapsible__container" style="max-height: 400px;">
  <div class="ui-pdp-description pl-24 pr-24 ui-pdp-collapsible--is-collapsed">
    <h2 class="ui-pdp-description__title">Descripción</h2>
    <p class="ui-pdp-description__content"> == $0
      "Con tu consola HMG-S-KAASA tendrás entretenimiento asegurado todos
      los días. Su tecnología fue creada para poner nuevos retos tanto a
      jugadores principiantes como expertos. "
      <br>
      <br>
      "Adaptada a tus necesidades"
      <br>
      "Guarda tus apps, fotos, videos y mucho más en el disco duro, que
      cuenta con una capacidad de 64 GB."
      <br>
      "Por otro lado, tiene puerto USB y salida HDMI, que permiten
      conectar accesorios y cargar la batería de tu control mientras
      juegas."
      <br>
      <br>
      "CÓMO DESCARGAR TU JUEGO."
      <br>
      "Al encender la consola, se debe conectar a internet e ingresar o
      crear tu usuario. En automático posterior a estos dos pasos el
      sistema te direccionará a la descarga del juego sin necesidad de
      ingresar ningún código."
      <br>
      <br>
      "NOTA:"
      <br>
      "Vas a poder reproducir música, ver tus películas y series favoritas
      a través de las aplicaciones descargables."
```


CONCLUSIONES

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.