



Centro de Investigación de Computo Instituto Politécnico Nacional



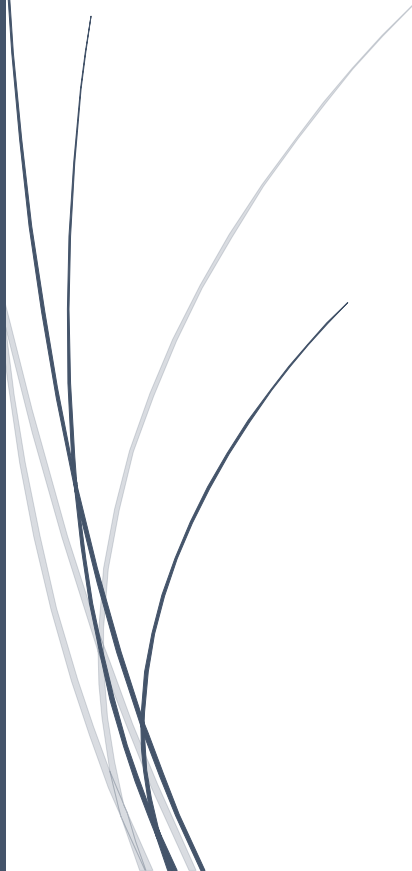
Cuso: Programación de Python en el ámbito Científico

Nombre: Daniela Téllez Jiménez

Correo: dany.tellez.jimenez98gmail.com

Profesor: Alan Badillo Salas

Fecha: Julio 2023



Introduccion

En los últimos años se ha tenido que manejar grandes cantidades de información. Algunos datos han sido extraídos de sitios web es por ello que empezó el uso de web scripting que es un conjunto de prácticas utilizadas para extraer automáticamente — o «scrapear» — datos de la web.

Justificación

En la siguiente practica se elaboró un scripting de la página de Sanbors, en donde se extrajo una lista de ciertos productos que maneja, así como su precio e imagen que nos podrá determinar que producto es el más caro, que tipo de productos venden o inclusive en un futuro poder hacer la comparación con otros precios que venden en otras tiendas.

Pasos a seguir

1. Se instalaron las librerías de Selenium la cual permite hacer el scripting en las páginas web y posteriormente Pandas que es para el manejo de grandes cantidades de datos a través de un DataFrame.

```
#Se importan Las librerías de selenium para hacer el scraping y también Las librerías de pandas para el manejo del Data Frame
from selenium import webdriver
import time
import pandas as pd
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
#Se selecciona en el navegador que se desea navegar y que cuando abra La página la extienda
driver = webdriver.Chrome()
driver.maximize_window()
```

2. Se accedió a la página de donde se extraerán los datos, en este caso en la página de Sanborns. Se debe primero buscar el nodo principal de donde se encuentra la información que se va a extraer para posteriormente de ahí irnos moviendo a los siguientes nodos.

```

#Accede a la página de sanborns y al nodo que se desea acceder junto con sus hijos
driver.get(" https://www.sanborns.com.mx/")
datos=driver.find_elements(By.XPATH, "//div[starts-with(@class, 'CardProduct_contDataCard')]")

#Se crean las listas donde se van almacenar los datos extraidos
precio_lista=[]
nombre_lista=[]
imagen_lista=[]

#Realiza la búsqueda de cada artículo que se encuentra en cada uno de los nodos
for nodo in datos:
    nombre=nodo.find_element(By.XPATH, "./h3[starts-with(@class, 'CardProduct_h4')]").text
    precio=nodo.find_element(By.XPATH, ".p[starts-with(@class, 'CardProduct_precio1')]").text
    imagen=nodo.find_element(By.XPATH, "../picture/img").get_attribute("src")

    #Valida que el nodo nombre o precio contenga información
    if nombre=='' or precio=='':
        continue

    #Se almacenan en las listas cada uno de los nodos extraidos
    precio_lista.append(precio)
    nombre_lista.append(nombre)
    imagen_lista.append(imagen)

```

3. Una vez obteniendo la información las guardamos en unas listas para poderlas guardar en un DataFrame y haya un mejor orden en los datos, así como un mejor manejo de la Data si se requiere utilizar para futuros trabajos.

```

#Se crean las listas donde se van almacenar los datos extraidos
precio_lista=[]
nombre_lista=[]
imagen_lista=[]

```

```

        continue

    #Se almacenan en las listas cada uno de los nodos extraidos
    precio_lista.append(precio)
    nombre_lista.append(nombre)
    imagen_lista.append(imagen)

#Se guarda en el data frame con la información de las listas
data={
    "nombre":nombre_lista,
    "precio":precio_lista,
    "imagen":nombre_lista
}

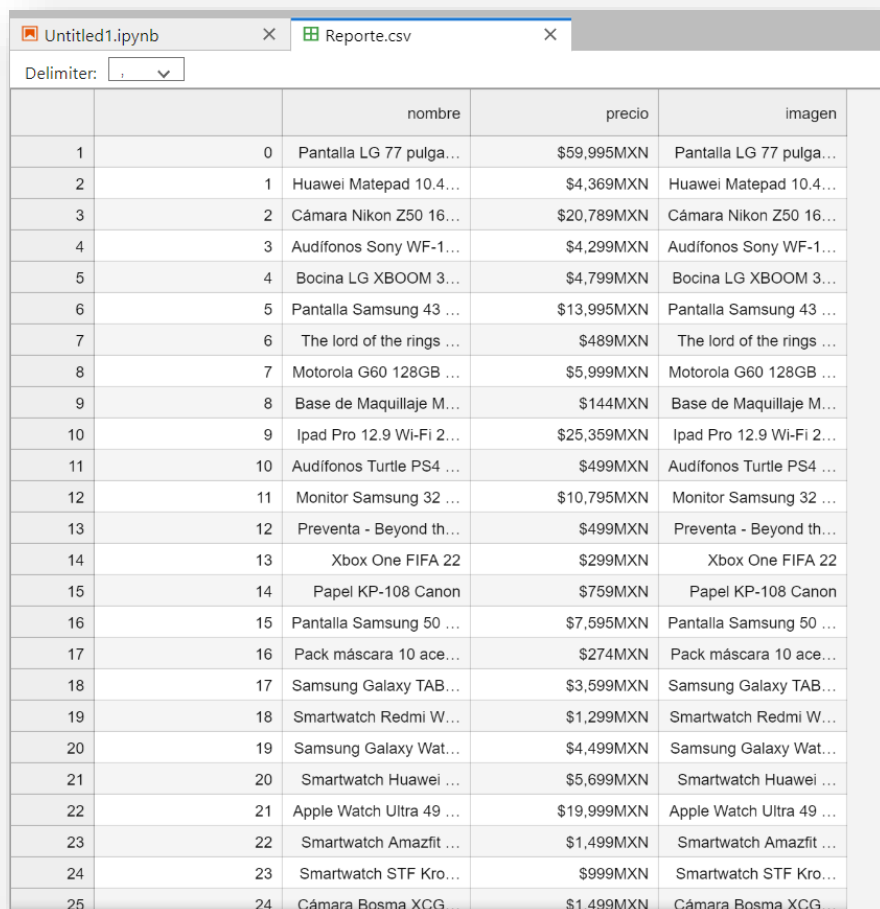
#Se crea el data frame
df = pd.DataFrame(data)

```

4. Pasamos el dataframe a un reporte csv para que podamos trabajar la Data en una data estructurada.

```
#Se crea el csv
df.to_csv("Reporte.csv")
#Se cierra el Selenium
driver.close()
```

Finalmente se genera el formato



The screenshot shows a Jupyter Notebook interface with two tabs: 'Untitled1.ipynb' and 'Reporte.csv'. The 'Reporte.csv' tab is active, displaying a table with 5 columns: an index column, an empty column, 'nombre', 'precio', and 'imagen'. The table contains 25 rows of product data, including items like 'Pantalla LG 77 pulga...', 'Huawei Matepad 10.4...', 'Cámara Nikon Z50 16...', etc.

		nombre	precio	imagen
1	0	Pantalla LG 77 pulga...	\$59,995MXN	Pantalla LG 77 pulga...
2	1	Huawei Matepad 10.4...	\$4,369MXN	Huawei Matepad 10.4...
3	2	Cámara Nikon Z50 16...	\$20,789MXN	Cámara Nikon Z50 16...
4	3	Audifonos Sony WF-1...	\$4,299MXN	Audifonos Sony WF-1...
5	4	Bocina LG XBOOM 3...	\$4,799MXN	Bocina LG XBOOM 3...
6	5	Pantalla Samsung 43 ...	\$13,995MXN	Pantalla Samsung 43 ...
7	6	The lord of the rings ...	\$489MXN	The lord of the rings ...
8	7	Motorola G60 128GB ...	\$5,999MXN	Motorola G60 128GB ...
9	8	Base de Maquillaje M...	\$144MXN	Base de Maquillaje M...
10	9	Ipad Pro 12.9 Wi-Fi 2...	\$25,359MXN	Ipad Pro 12.9 Wi-Fi 2...
11	10	Audifonos Turtle PS4 ...	\$499MXN	Audifonos Turtle PS4 ...
12	11	Monitor Samsung 32 ...	\$10,795MXN	Monitor Samsung 32 ...
13	12	Preventa - Beyond th...	\$499MXN	Preventa - Beyond th...
14	13	Xbox One FIFA 22	\$299MXN	Xbox One FIFA 22
15	14	Papel KP-108 Canon	\$759MXN	Papel KP-108 Canon
16	15	Pantalla Samsung 50 ...	\$7,595MXN	Pantalla Samsung 50 ...
17	16	Pack máscara 10 ace...	\$274MXN	Pack máscara 10 ace...
18	17	Samsung Galaxy TAB...	\$3,599MXN	Samsung Galaxy TAB...
19	18	Smartwatch Redmi W...	\$1,299MXN	Smartwatch Redmi W...
20	19	Samsung Galaxy Wat...	\$4,499MXN	Samsung Galaxy Wat...
21	20	Smartwatch Huawei ...	\$5,699MXN	Smartwatch Huawei ...
22	21	Apple Watch Ultra 49 ...	\$19,999MXN	Apple Watch Ultra 49 ...
23	22	Smartwatch Amazfit ...	\$1,499MXN	Smartwatch Amazfit ...
24	23	Smartwatch STF Kro...	\$999MXN	Smartwatch STF Kro...
25	24	Cámara Bosma XCG...	\$1,499MXN	Cámara Bosma XCG...

Conclusión

Este tipo de prácticas ayuda a las empresas con los procesos basados en datos, desde el seguimiento de las marcas y las comparaciones de precios actualizadas hasta la realización de valiosos estudios de mercado. Son algunos de los más comunes.