¿Cómo elaborar un Webscraper?

En este documento se explicará paso a paso cómo realizar un Websrapper basándonos en el que realizamos como proyecto final.

- Paso 1: Traer todas las librerías necesarias, las cuales mostraremos a continuación.

```
import pandas as pd
import pandas.io.sql as sqlio

from bs4 import BeautifulSoup
from urllib.request import urlopen
import urllib.request
import requests
import time
from multiprocessing import Process, Queue, Pool
import threading
import sys
import numpy as np
import re

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.options import Options
import pandasql as ps
from IPython.display import display,HTML
from datetime import date
from datetime import datetime
import matplotlib.pyplot as plt
```

- Paso 2: Ya instalado un web driver verificar la ruta y asignarla.

```
    path ="C:\webdriver\chromedriver.exe" # carga del web driver (asignar ruta donde se encuentra el driver)
    driver=webdriver.Chrome(path)

    url="https://www.farmaciasanpablo.com.mx/search/" + marca
    driver.get(url) # instruccion de obtener url parametrizada
```

- Paso 3: Hacer las siguientes funciones:
  i) Hacer una función por cada sitio que quieras visitar, por ejemplo, si visitarás tres sitios harás tres funciones.
  ii) Cada función recibirá como parámetro el producto o marca a buscar.

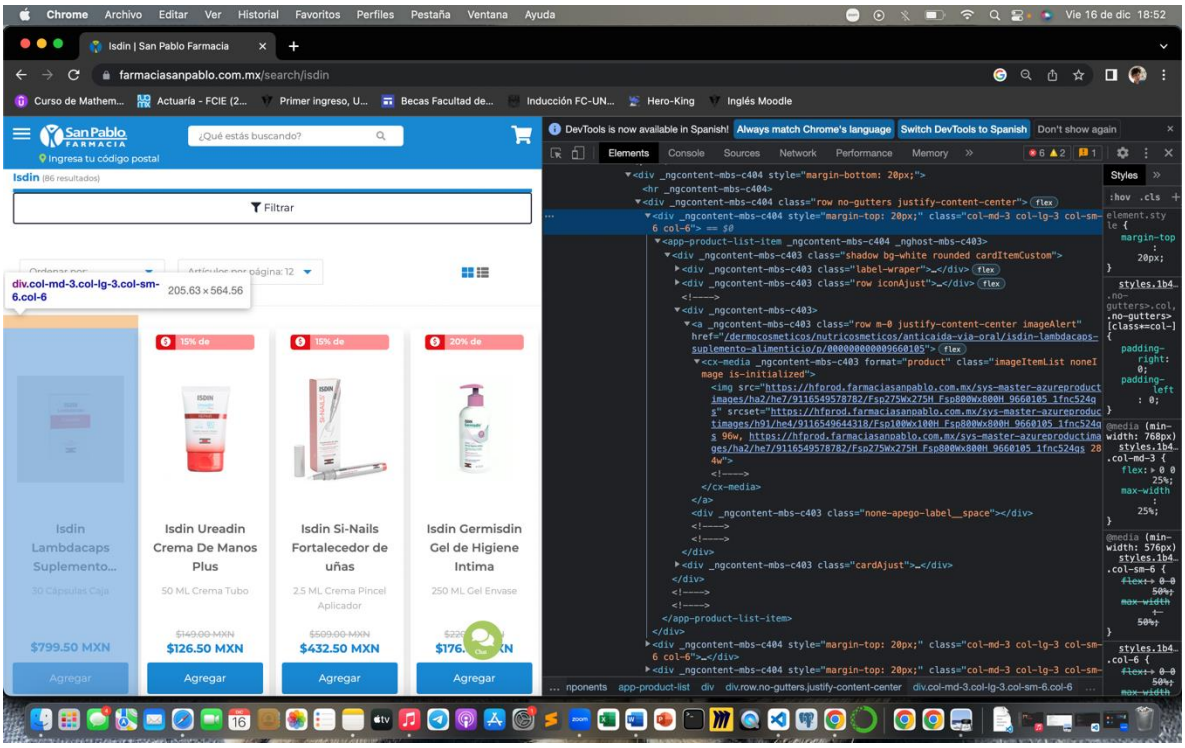
```
def san_pablo(marca):
```

  iii) En cada una de las funciones ser utilizará el driver. Asignar la ruta en la que se encuentra el Web driver; conseguir la URL parametrizada de las búsquedas del sitio, es decir qué parte de la URL es fija cada vez que se hace una búsqueda; con el driver, método get y URL parametrizada se hace la búsqueda de manera automatizada; dormir al equipo (función time.sleep()).

```
    url="https://www.farmaciasanpablo.com.mx/search/" + marca
    driver.get(url) # instruccion de obtener url parametrizada

    time.sleep(12) # dormir equipo para carga de pagina
```

  iv) Buscar clases de cómo está organizado el sitio que quieres utilizar (usando la función de inspeccionar en Chrome), este será el nombre de la clase. Al realizar la búsqueda con el driver hay que buscar la clase que contenga a toda la información de cada producto en una página




```
e busca clase que contiene todos los items mostrados en la pagina
 productos=driver.find_elements_by_class_name("col-md-3.col-lg-3.col-sm-6.col-6")

 lista_nombres=[]
 for i in range(0,len(productos)):
     try:
         lista_nombres.append(productos[i].find_elements_by_class_name("nameProduct")[0].text)
     except:
         lista_nombres.append(np.nan)

signacion de precios recientes (con descuento)
 lista_precios=[]
 for i in range(0,len(productos)):
     try:
         lista_precios.append(productos[i].find_elements_by_class_name("col-md-11.col-lg-11.col-sm-11.col-11.price")[0].text)
     except:
         lista_precios.append(np.nan)


ecio sin descuento
 lista_precios_sin_descuento=[]
 for i in range(0,len(productos)):
     try:
         lista_precios_sin_descuento.append(productos[i].find_elements_by_class_name("discount")[0].text)
     except:
         lista_precios_sin_descuento.append(np.nan)
```

v) Declarar una lista vacía por cada parámetro que quieras buscar, después con un ciclo for se llenará cada lista vacía, como nombre del parámetro se utiliza el resultado de la inspección en Chrome (Mirar imagen anterior).

```
lista_nombres=[]
```

vi) Realizar un DataFrame por cada página visitada y declarar una columna por cada lista (parámetro) que se buscó a través del driver.

```
# creacion de data frame
    today = date.today()

    df_sp =pd.DataFrame(columns=["NOMBRE","MARCA","PRECIO","PRECIO_SIN_DESCUENTO","DISPONIBILIDAD","AUTOSERVICIO","FECHA"])
    df_sp["NOMBRE"] = lista_nombres
    df_sp["MARCA"] = marca
    df_sp["PRECIO"] = lista_precios
    df_sp["PRECIO_SIN_DESCUENTO"]=lista_precios_sin_descuento
    df_sp["DISPONIBILIDAD"] = "Desconocido"
    df_sp["AUTOSERVICIO"] = "www.farmaciasanpablo.com.mx"
    df_sp["FECHA"] = str(today)

    df_sp.PRECIO = df_sp.PRECIO.str.replace("MXN","")
    df_sp.PRECIO = df_sp.PRECIO.str.replace("$","")
    df_sp.PRECIO = df_sp.PRECIO.str.replace(",","")
    df_sp.PRECIO = df_sp.PRECIO.astype(float) # cast de datos

    try:
        df_sp.PRECIO_SIN_DESCUENTO = df_sp.PRECIO_SIN_DESCUENTO.str.replace("MXN","")
        df_sp.PRECIO_SIN_DESCUENTO = df_sp.PRECIO_SIN_DESCUENTO.str.replace("$","")
        df_sp.PRECIO_SIN_DESCUENTO= df_sp.PRECIO_SIN_DESCUENTO.str.replace(",","")
        df_sp.PRECIO_SIN_DESCUENTO= df_sp.PRECIO_SIN_DESCUENTO.astype(float) # cast de datos
    except:
        pass

    driver.quit()
    return df_sp
```

vii) A partir de expresiones regulares, limpiar el formato de los datos que se obtuvieron, se busca darles un formato que facilite su tratamiento (Mirar la imagen anterior).

El paso 3 debe ser repetido por cada página que se visita.

- Paso 4: Hacer las búsquedas necesarias por cada marca y producto deseado, por ejemplo, seleccionar marcas y buscarlas en cada uno de los sitios.

```
busqueda1 = san_pablo("isdin")
time.sleep(10)
busqueda1
```

```
C:\Users\mgonz\AppData\Local\Temp\ipykernel_4724\3335917239.py:50: FutureWarning: The default value of regex will change from T
rue to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings wh
en regex=True.
  df_sp.PRECIO = df_sp.PRECIO.str.replace("$","")
C:\Users\mgonz\AppData\Local\Temp\ipykernel_4724\3335917239.py:56: FutureWarning: The default value of regex will change from T
rue to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings wh
en regex=True.
  df_sp.PRECIO_SIN_DESCUENTO = df_sp.PRECIO_SIN_DESCUENTO.str.replace("$","")
```

	NOMBRE	MARCA	PRECIO	PRECIO_SIN_DESCUENTO	DISPONIBILIDAD	AUTOSERVICIO	FECHA
0	Isdin Lambdacaps Suplemento Alimenticio	isdin	799.5	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
1	Isdin Ureadin Crema De Manos Plus	isdin	126.5	149.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
2	Isdin Si-Nails Fortalecedor de uñas	isdin	432.5	509.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
3	Isdin Germisdin Gel de Higiene Intima	isdin	176.0	220.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
4	Isdin Fusion Water Urban Protector Solar	isdin	623.5	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
5	Isdin Instant Flash Suero Antiedad Efec...	isdin	303.0	357.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
6	Isdin Hialuronic Booster Serüm Antiedad...	isdin	357.0	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
7	Isdin Fotoprotector FPS50+ Gel Cream Pa...	isdin	645.0	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
8	Isdin Fusion Water Fotoprotector Facial...	isdin	580.0	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
9	Isdin Germisdin Aloe Vera Gel De Baño P...	isdin	290.5	363.5	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
10	Isdin Foto Ultra Age Repair Fusion Wate...	isdin	599.0	705.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
11	Isdin FLAVO-C MELATONIN SÉRUM REPARADOR...	isdin	686.0	857.5	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16

- Paso 5: Después de hacer las búsquedas deseadas, podemos hacer un DataFrame que coleccione a todas las búsquedas hechas por sitio web, se puede usar la función para concatenar DataFrame para formar uno solo (pd.concat). En este mismo paso, se pueden tratar con excepciones los segmentos de datos vacíos.

```
df_farmazone_final = pd.concat([busqueda6, busqueda7, busqueda8, busqueda9, busqueda10])
df_farmazone_final = df_farmazone_final[df_farmazone_final.PRECIO!=""]
```

- Paso 6: Después de hacer todos esos DataFrames, volver a usar la función para concatenar y así hacer un último DataFrame que contenga todo.

```
df_final = pd.concat([df_san_pablo_final, df_farmazone_final, df_farmalisto_final])
```

```
df_final
```

	NOMBRE	MARCA	PRECIO	PRECIO_SIN_DESCUENTO	DISPONIBILIDAD	AUTOSERVICIO	FECHA
0	Isdin Lambdacaps Suplemento Alimenticio	isdin	799.50	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
1	Isdin Ureadin Crema De Manos Plus	isdin	126.50	149.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
2	Isdin Si-Nails Fortalecedor de uñas	isdin	432.50	509.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
3	Isdin Germisdin Gel de Higiene Intima	isdin	176.00	220.0	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
4	Isdin Fusion Water Urban Protector Solar	isdin	623.50	NaN	Desconocido	www.farmaciasanpablo.com.mx	2022-12-16
...	...	...	...	...	...	...	...
45	Antitranspirante Nívea Men Stress Protect Aero...	nivea	64.26	NaN	Desconocido	www.farmalisto.com.mx	2022-12-16
46	Antitranspirante Nivea Dry Impact Roll-On Con ...	nivea	27.84	NaN	Desconocido	www.farmalisto.com.mx	2022-12-16
47	Antitranspirante Nivea Aclarado 48H Roll-On Co...	nivea	34.00	NaN	Desconocido	www.farmalisto.com.mx	2022-12-16
48	Antitranspirante Nivea Aclarado Natural Roll-O...	nivea	33.00	NaN	Desconocido	www.farmalisto.com.mx	2022-12-16

- Paso 7: Declarar un archivo de Excel (.xlsx) que exporte a una hoja de Excel nuestro DataFrame.

```
#Documento excel con los datos del dataframe final
df_final.to_excel("df_final.xlsx",index=False)
```

- Nota 1: Es conveniente que los DataFrames contengan la misma cantidad de parámetros.
- Paso 8: Usar la librería plot para graficar los parámetros de cada búsqueda, se pueden hacer nuevas consultas por búsqueda para especificar cada uno de los gráficos.

```
#Precios de la marca isdin
precios_isdin = df_final[df_final.MARCA=="isdin"]["PRECIO"].plot(kind="hist", color="purple")
```