Universidad Nacional Autónoma de México

Facultad de ciencias

Manejo de datos

Profesor(es):

Jessica Santizo Galicia

Sergio Alejandro Chávez Molotla

Integrantes:

Gónzalez Robles Sofía Quetzalli

Manríquez Rangel Armando Daniel

Mariano Martínez Kevin

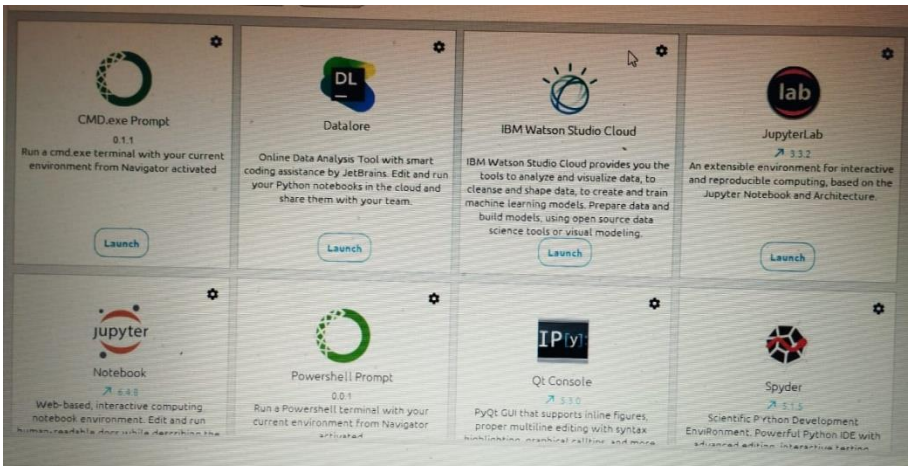
Serralde Salinas Alejandro

Semestre 2023-1

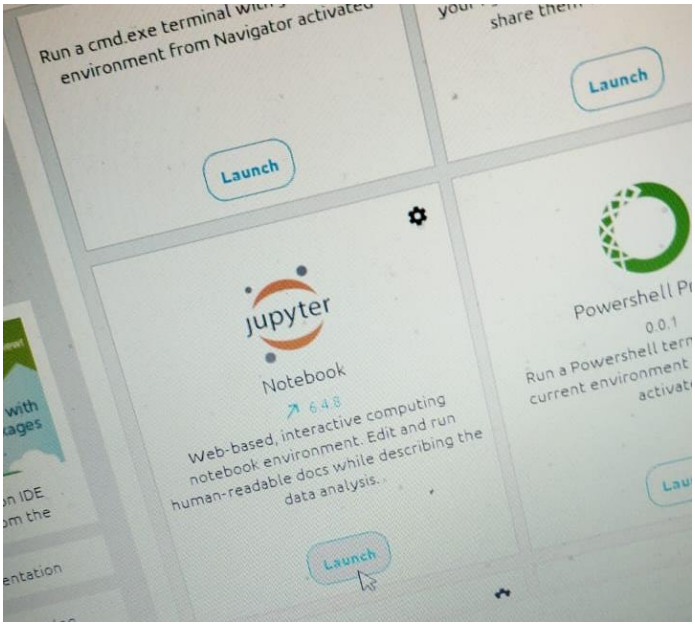
Weebscrapper.pdf

Bueno a continuación vamos a dar los pasos para la realización de un webscreapper sabiendo conocimientos de Python, lo normal por ello, a continuación, daremos de una manera detallada manera para que todos lo podamos entender:

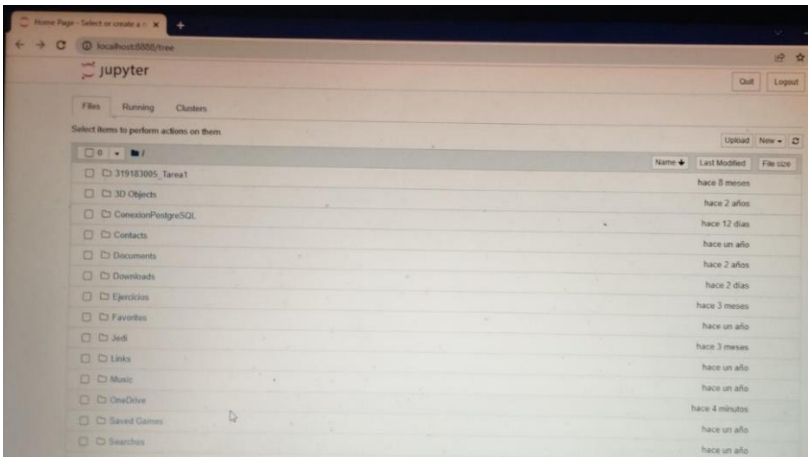
- Una vez con Anaconda instalado, en el buscador buscamos Anaconda Navigator, nos abrirá esto



- Abrimos el jupyter notebook para un mejor manejo



- Y nos va a mandar aquí



- Ya con ello creamos un nuevo proyecto en nuestro jupyter notebook
- Primero importamos las librerías necesarias:

```
[ ]  # Librerías a utilizar:
     import pandas as pd
     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 random_user_agent.user_agent import UserAgent
     #from random_user_agent.params import SoftwareName, OperatingSystem
     from selenium import webdriver
     from selenium.webdriver.common.keys import Keys
     from selenium.webdriver.common.by import By
     #from fake_useragent import UserAgent
     from selenium.webdriver.chrome.options import Options
     import pandasql as ps
     from IPython.display import display,HTML
     import matplotlib.pyplot as plt
```

*Nota recordando un poco de lo que hace algunas librerías sabemos que

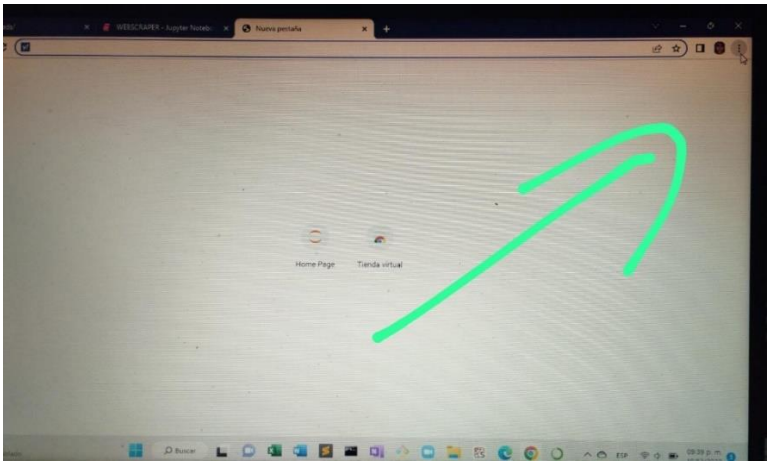
- Pandas nos ayudan a la facilitación de leer y escribir datos, filtrar mas rápido lo que queremos, unir datos entre otras cosas
- Numpy un mejor uso de cálculos matemáticos y arrays
- Matplotlib para el uso de graficas y visualización en este caso para comparar entre otras y Para la parte de graficas descargamos: Selenium, Pandasql e importamos Beautiful Soup, pandas, Requests

-Una vez con lo necesario haremos primero definir nuestra función de la primera tienda a elegir en nuestro caso fue de Zara

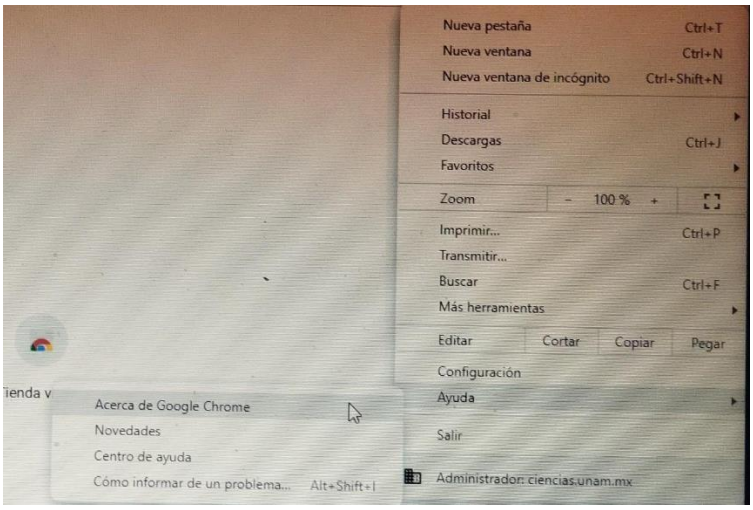
▾ ZARA

```
[ ]  def Zara_proyecto(producto):
         """
         Función que hace WEB SCRAPPING EN LA PÁGINA DE ZARA
         """
```

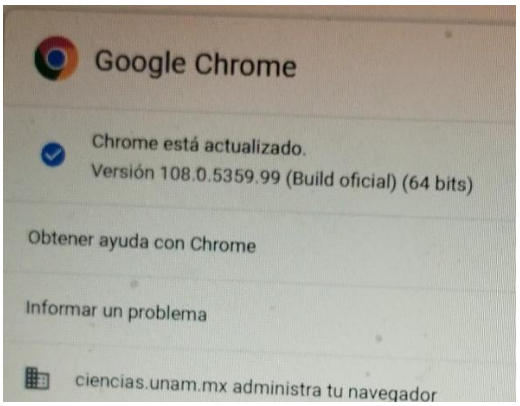
- Una vez con esto, nos falta instalar el web driver, abriremos nuestro Google Chrome y le damos en especificaciones para conocer nuestra versión.



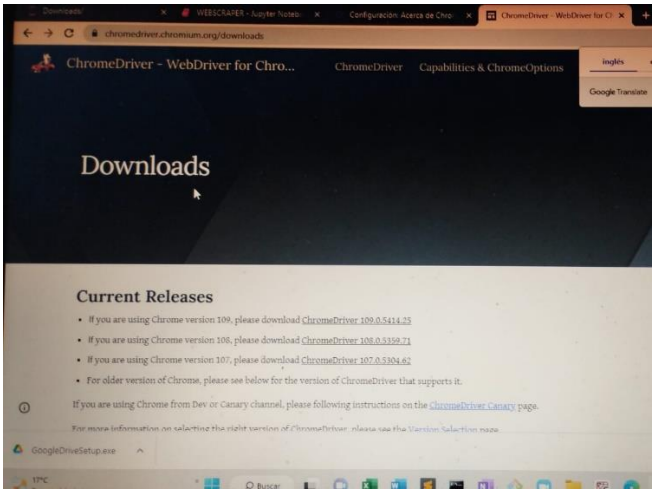
- Le dan en ayuda y en la ventana que despliega le damos en acerca de Google Chrome



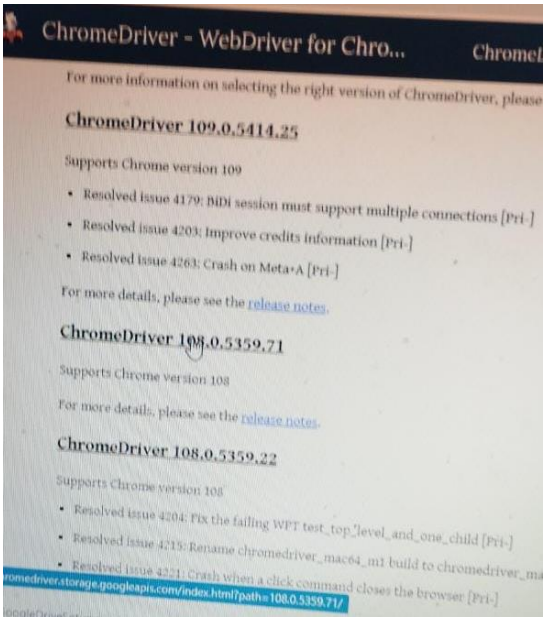
- Ya con las especificaciones y versión de nuestro Chrome (Ejemplo)



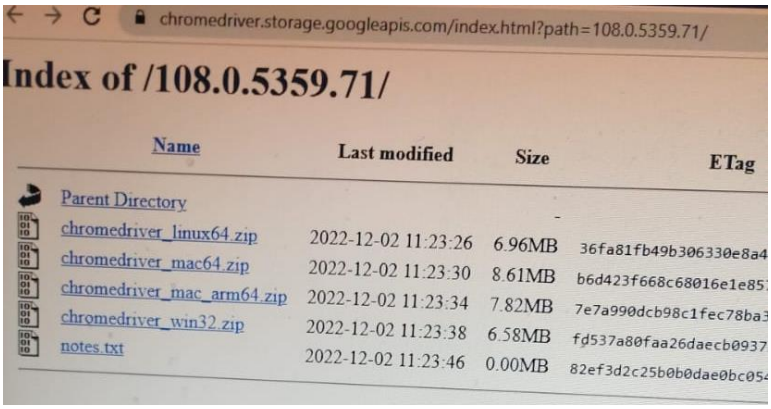
- Buscamos en el buscador Chrome driver



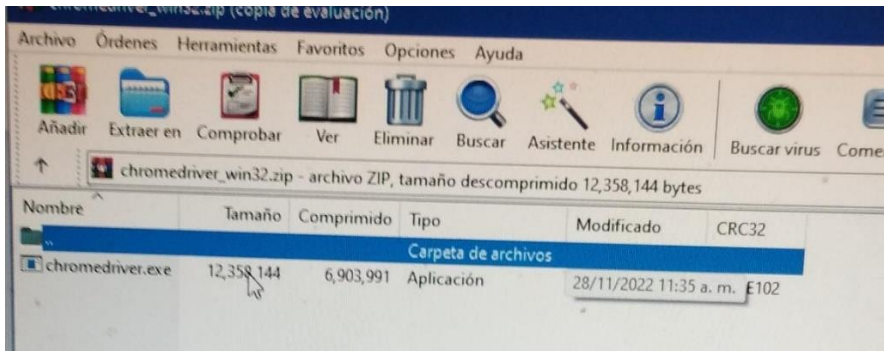
- Si no aparece nuestra versión ponemos la que más se acerca



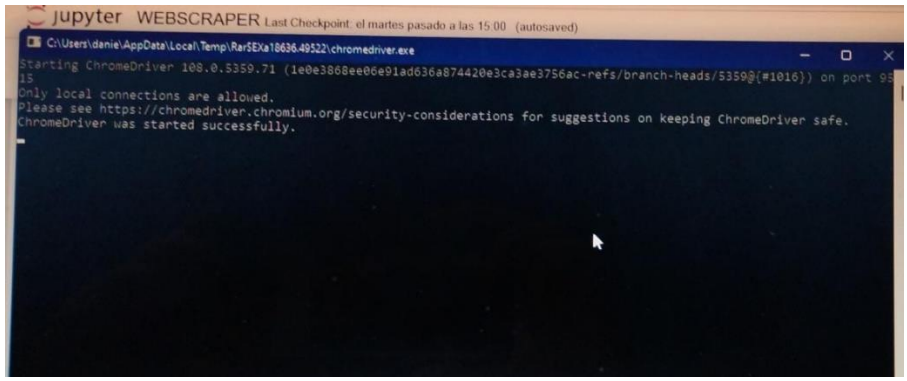
- Escogemos la opción de nuestro sistema operativo



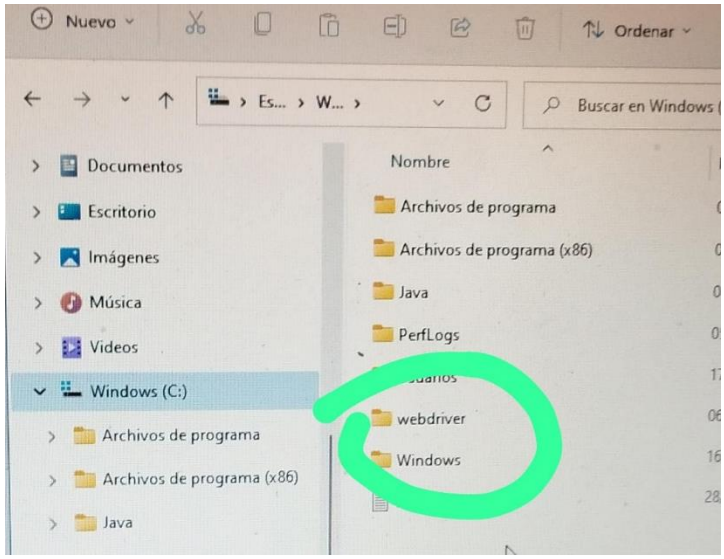
- Lo descargamos y descomprimimos



- Estará listo cuando nos aparezca esta ventana



- Para esto del webdriver abrimos una carpeta de fácil acceso en nuestro caso la llamamos webdriver



- Continuando con el código, ponemos la ruta path = a nuestra ubicación del Chrome driver

ZARA

```
[ ]  def Zara_proyecto(producto):
        """
        Función que hace WEB SCRAPPING EN LA PÁGINA DE ZARA
        """
        path = "C:\webdriver\chromedriver.exe"
        driver=webdriver.Chrome(path)
        time.sleep(8)
```

- Después de buscar nuestra tienda copiamos el url hasta antes del producto a nuestra elección y nuevamente dormimos para cargar por completo la pagina

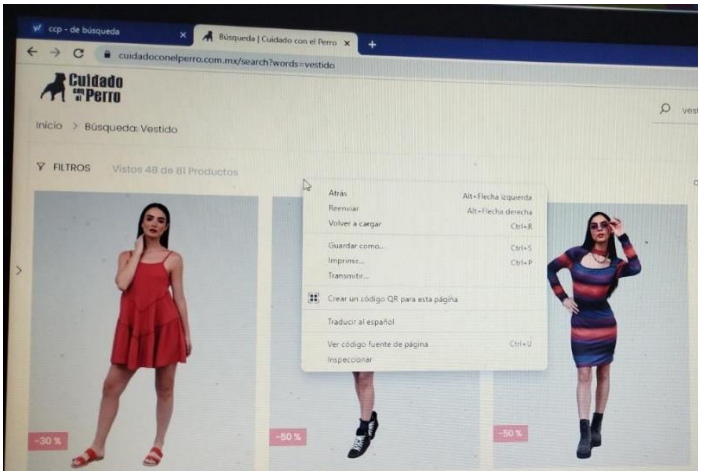
```
        ### URL de la pagina web al momento de realizar una busqueda de un producto
        url = "https://www.zara.com/mx/es/search?searchTerm="+producto+"&section=WOMAN"
        driver.get(url)
        ###La dormimos, de lo contrario no es posible obtener la informacion
        time.sleep(8)
```

- El paso siguiente es hacer la clase para los productos elegidos y repetimos el proceso para que se cargue bien la pagina

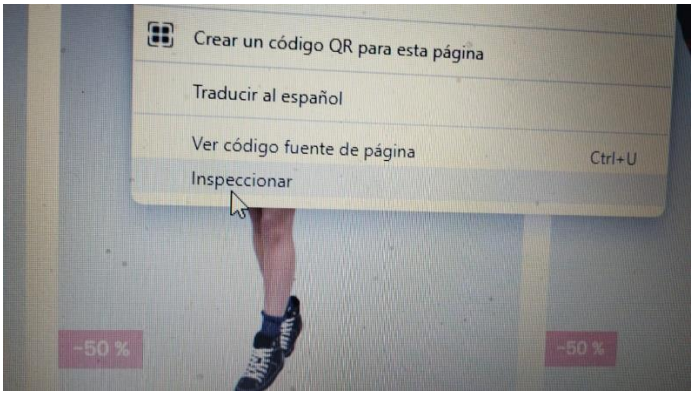
```
        """
        Desde aquí se buscan las clases para los apartados en cuestion
        """
        ###El que abarca todo
        productos= driver.find_elements_by_class_name("product-grid-product__info-wrapper")
        time.sleep(8)
```

- Ahora haremos la extracción de las clases necesarias para los apartados de precios hora entre otros

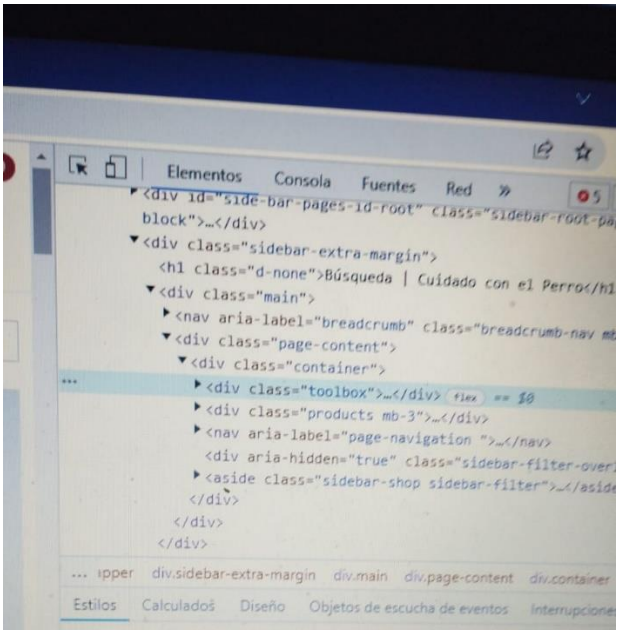
-Van a la página que piensan hacer y en cualquier parte de la pantalla das click derecho aparecerá lo siguiente



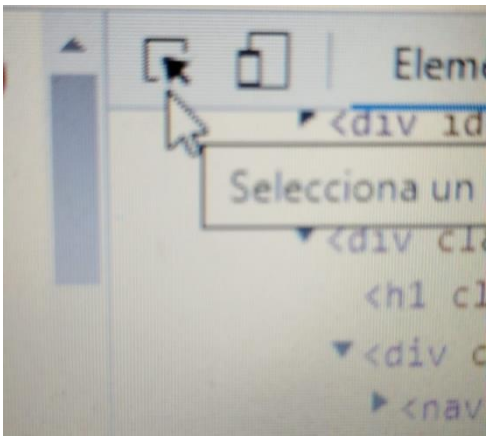
- Le damos en dónde dice inspeccionar



- Despliega lo siguiente en la misma pagina

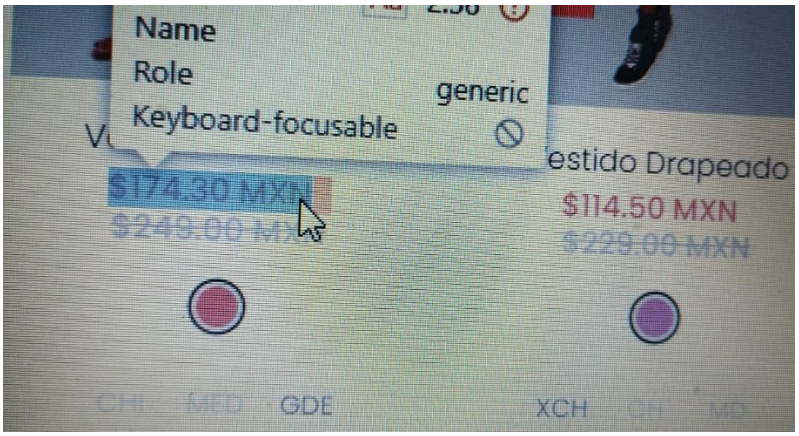


- Le picamos en la lupa para fácil manejo

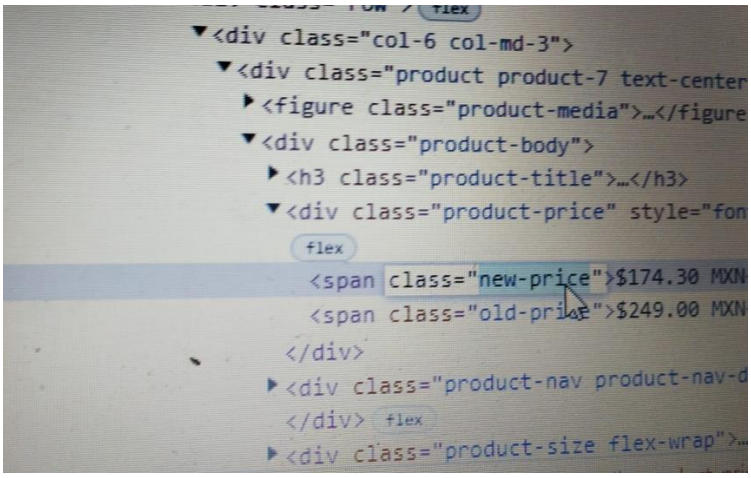


- Entonces seleccionas el apartado donde está la información que buscas

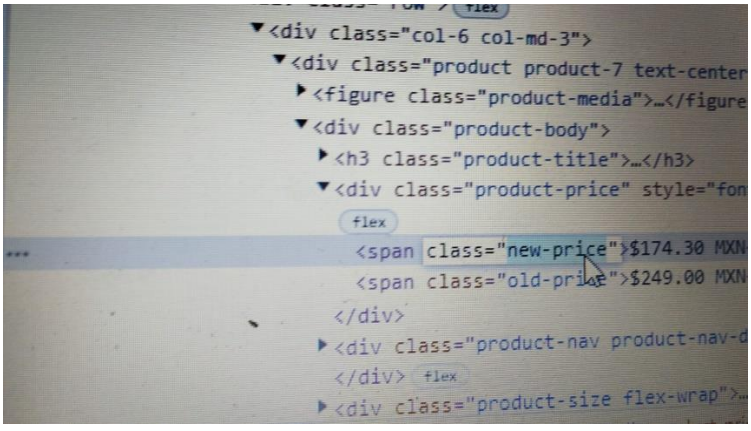
-Cómo por ejemplo nuevo precio



- En el panel se subraya la clase, ya que será la que nos ayudara la copiamos



- Pegamos aquí



- Ya con todas los clases solo hacemos un ciclo para tener todos los productos del articulo que queramo

```
[ ]
        ###Como extra encontramos como obtener el URL y lo plasmamos aquí, se guarda en una lista
        lista_urls=list()
        for i in range(len(productos)):
            try:
                lista_urls.append(productos[i].find_element_by_tag_name("a").get_attribute("href"))
            except:
                lista_urls.append(np.nan)
        time.sleep(8)

        ###Lista donde guardamos el nombre del producto
        lista_nombres=list()
        for i in range(len(productos)):
            try:
                lista_nombres.append(productos[i].find_elements_by_tag_name("a")[0].text)
            except:
                lista_nombres.append(np.nan)

        time.sleep(8)
```

- Ahora haremos el paso para guardar los datos de la pagina los cuales se quedaran en un futuro dataframe obtenidos de las respectivas clases

```
[ ]
        time.sleep(8)
        ###Lista donde guardamos el precio, se maneja como precio final el precio ya con descuento o cuando no cuenta con descuento
        ###El precio anterior es cuando si hay descuento, es el precio antes del descuento
        ###En caso de no tener se coloca un NaN
        lista_precios=list()
        lista_promos=list()
        for i in range(len(productos)):
            try:
                lista_precios.append(productos[i].find_elements_by_class_name("price-current__amount")[0].text)
            except:
                lista_precios.append(np.nan)
            try:
                lista_promos.append(productos[i].find_elements_by_class_name("price-old__amount")[0].text)
            except:
                lista_promos.append(np.nan)
        ###Se hace el DataFrame con sus columnas
        df_zara = pd.DataFrame(columns=["Nombre","URL","PrecioFinal","PrecioAnterior"])
```

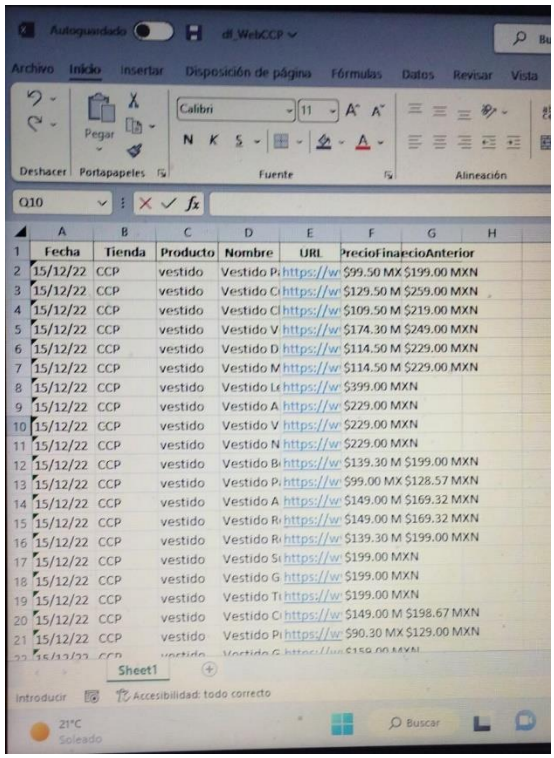
- Obtenemos el data frame con todos nuestros datos requeridos y los nombramos para un mejor manejo

```
    ###Se llena la tabla con la información obtenida
    df_zara["Nombre"]= lista_nombres
    df_zara["URL"]= lista_urls
    df_zara["PrecioFinal"]= lista_precios
    df_zara["PrecioAnterior"]= lista_promos
    df_zara["Tienda"]="ZARA"
    df_zara["Producto"]= producto
    df_zara["Fecha"]= time.strftime("%d/%m/%y")
    ###Devolvemos el Data Frame
    df_zara = df_zara[["Fecha","Tienda","Producto","Nombre","URL","PrecioFinal","PrecioAnterior"]]
    driver.quit()
    return df_zara
```

- Pasamos a Excel, en este paso es, en el cual llamaremos como queramos nuestro .xls con nuestros información obtenida recordando lo nombramos con comillas “”

```
[ ]  ###Pasamos a el excel
     df_ccp.to_excel("df_WebCCP.xlsx",index=False)
```

- Los productos quedaran juntos así que en este paso es en el que nuestro dataframe nos dará el total de Vestidos, Chamarras y Pantalones.



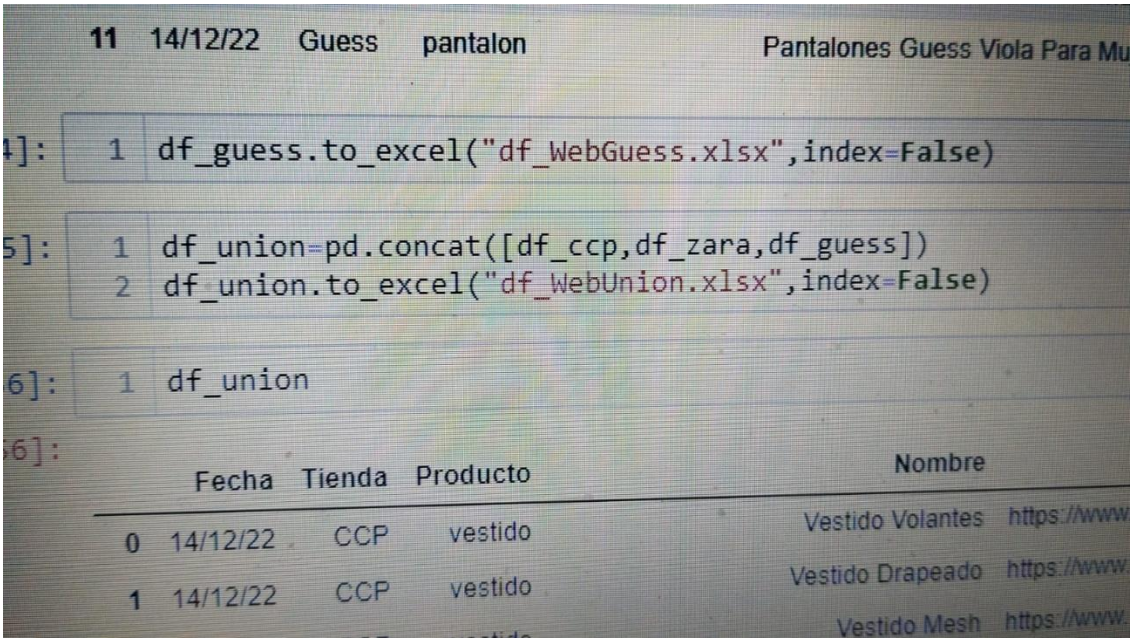
- Mostramos el data frame, como lo vimos en clase este solo alcanza a mostrar porcas columnas ya que el espacio en el notebook es reducido pero en la parte de abajo se alcanza a ver cuantas columnas y filas son en este caso 90 productos de zara y 7 que son nuestros datos (fecha, url, producto etc) sin importar

```
[ ]  ###Mostramos el resultado
     df_zara
```

	Fecha	Tienda	Producto	Nombre	URL	PrecioFinal	PrecioAnterior
0	14/12/22	ZARA	vestido	VESTIDO SKATER PUNTO	https://www.zara.com/mx/es/vestido-skater-punt...	1.199,00 MXN	NaN
1	14/12/22	ZARA	vestido	VESTIDO PLISADO CINTURÓN	https://www.zara.com/mx/es/vestido-plisado-cin...	699,00 MXN	1.299,00 MXN
2	14/12/22	ZARA	vestido	VESTIDO CRUZADO SATINADO	https://www.zara.com/mx/es/vestido-cruzado-sat...	899,00 MXN	NaN
3	14/12/22	ZARA	vestido	VESTIDO MIDI JACQUARD	https://www.zara.com/mx/es/vestido-midi-jacqua...	1.299,00 MXN	NaN
4	14/12/22	ZARA	vestido	VESTIDO CORTO ESTAMPADO	https://www.zara.com/mx/es/vestido-corto-estam...	1.199,00 MXN	NaN
...	...	...	...	...	...	...	...
25	14/12/22	ZARA	pantalon	PANTALÓN FLARE TERCIOPELO	https://www.zara.com/mx/es/pantal%C3%B3n-flare...	1.299,00 MXN	NaN
26	14/12/22	ZARA	pantalon	PANTALÓN ZW THE MARINE STRAIGHT EFECTO PIEL	https://www.zara.com/mx/es/pantal%C3%B3n-zw-th...	899,00 MXN	NaN
27	14/12/22	ZARA	pantalon	PANTALÓN SATINADO RECTO	https://www.zara.com/mx/es/pantal%C3%B3n-satin...	1.299,00 MXN	NaN
28	14/12/22	ZARA	pantalon	PANTALÓN JOGGER SATINADO EFECTO ARRUGADO	https://www.zara.com/mx/es/pantal%C3%B3n-jogge...	749,00 MXN	NaN
29	14/12/22	ZARA	pantalon	PANTALÓN ABERTURAS LATERALES	https://www.zara.com/mx/es/pantal%C3%B3n-abert...	899,00 MXN	NaN

90 rows × 7 columns

- Un punto importante es el de concat ya que nuestros tres dataframes los unirá en uno solo y de igual forma ese es el que haremos un .xls



```
11  14/12/22  Guess  pantalon  Pantalones Guess Viola Para Mu

4]:  1  df_guess.to_excel("df_WebGuess.xlsx",index=False)

5]:  1  df_union=pd.concat([df_ccp,df_zara,df_guess])
     2  df_union.to_excel("df_WebUnion.xlsx",index=False)

6]:  1  df_union

6]:
```

	Fecha	Tienda	Producto	Nombre	
0	14/12/22	CCP	vestido	Vestido Volantes	https://www
1	14/12/22	CCP	vestido	Vestido Drapeado	https://www
				Vestido Mesh	https://www

- Otro punto que hicimos son las consultas sql aprendidas en las clases en este caso un ejemplo del url de los vestidos de Zara con un precio menor a 1000 pesos

```
[ ]  ###Consulta donde muestra URL de los vestidos con precio final menor a 1000 pesos
     ps.sqldf("select  URl from df_zara where (Producto = 'vestido') and PrecioFinal < 1000")
```

	URL
0	https://www.zara.com/mx/es/vestido-plisado-cin...
1	https://www.zara.com/mx/es/vestido-cruzado-sat...
2	https://www.zara.com/mx/es/vestido-camisero-te...
3	https://www.zara.com/mx/es/vestido-corto-satin...
4	https://www.zara.com/mx/es/vestido-soft-joyas-...
5	https://www.zara.com/mx/es/vestido-corto-satin...
6	https://www.zara.com/mx/es/vestido-estampado-v...
7	https://www.zara.com/mx/es/vestido-terciopelo-...
8	https://www.zara.com/mx/es/vestido-corto-terci...

- Por ultimo hacemos unas graficas de la comparación de nuestros tres sitios web para ver los diferentes precios de nuestros productos en este caso el de los vestidos

Tutorial Gráficas:

1. Primero revisamos nuestro Excel de las diferentes tiendas que se genera después de correr nuestro código:

	Fecha	Tienda	Producto	Nombre	URL	PrecioFinal	PrecioAnterior
2	14/12/22	Guess	vestido	Vestido G	https://w	$1,036.00	$2,590.00
3	14/12/22	Guess	vestido	Vestido G	https://w	$956.00	$2,390.00
4	14/12/22	Guess	vestido	Vestido G	https://w	$1,556.00	$3,890.00
5	14/12/22	Guess	vestido	Vestido G	https://w	$2,590.00	
6	14/12/22	Guess	vestido	Vestido G	https://w	$756.00	$1,890.00
7	14/12/22	Guess	vestido	Vestido G	https://w	$1,036.00	$2,590.00
8	14/12/22	Guess	vestido	Vestido G	https://w	$956.00	$2,390.00
9	14/12/22	Guess	vestido	Vestido G	https://w	$1,156.00	$2,890.00
10	14/12/22	Guess	vestido	Vestido G	https://w	$1,156.00	$2,890.00
11	14/12/22	Guess	vestido	Vestido G	https://w	$1,156.00	$2,890.00
12	14/12/22	Guess	vestido	Vestido G	https://w	$1,556.00	$3,890.00
13	14/12/22	Guess	vestido	Vestido G	https://w	$2,790.00	
14	14/12/22	Guess	chamarra	Chamarra	https://w	$3,190.00	
15	14/12/22	Guess	chamarra	Chamarra	https://w	$4,290.00	
16	14/12/22	Guess	chamarra	Chamarra	https://w	$4,490.00	
17	14/12/22	Guess	chamarra	Chamarra	https://w	$3,790.00	
18	14/12/22	Guess	chamarra	Chamarra	https://w	$1,516.00	$3,790.00
19	14/12/22	Guess	chamarra	Chamarra	https://w	$4,090.00	
20	14/12/22	Guess	chamarra	Chamarra	https://w	$5,990.00	

2. Con las funciones de nuestra hoja electrónica consultamos los valores de los productos más costosos y más económicos, y llevamos esos datos a mi Jupyter.

```
fig=plt.plot([90,229,249], l
fig=plt.plot([756,1516,876]
fig=plt.plot([899,899,449] ,
```

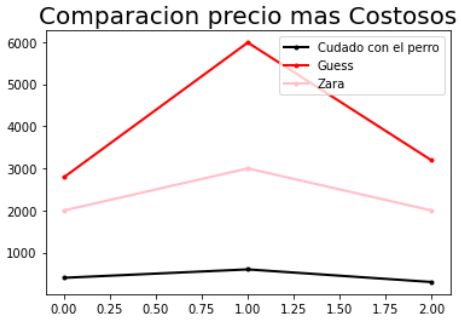
```
fig=plt.plot([399,599,299], l
fig=plt.plot([2790,5990,3190]
fig=plt.plot([1999,2999,1999]
```

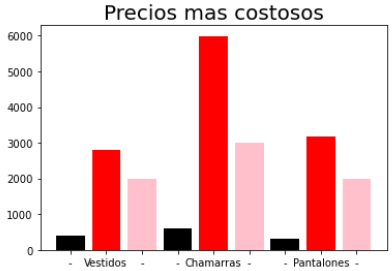
3. Después hacemos el código de nuestras gráficas, importando la librería matplotlib.pyplot y apoyándonos en el documento “Visualización” que se checo en clase para poder definir el estilo de nuestras gráficas, tamaño, marcadores, colores y ponerle etiquetas.

```
fig=plt.plot([399,599,299], linestyle="-",linewidth=2,marker="o",markersize=3 ,color="black",label="Cudado con el perro")
fig=plt.plot([2790,5990,3190] , linestyle="-",linewidth=2,marker="o",markersize=3 ,color="red",label="Guess")
fig=plt.plot([1999,2999,1999] , linestyle="-",linewidth=2,marker="o",markersize=3 ,color="pink",label="Zara")
plt.title("Comparacion precio mas Costosos",fontsize=20)
plt.legend()
plt.show()
```

```
fig, ax = plt.subplots()
plt.bar(range(9), [399,2790, 1999, 599, 5990, 2999, 299, 3190, 1999], color=["black","red","pink","black","red","pink","black","r
plt.title("Precios mas costosos",fontsize=20)
names = [ "-","Vestidos","-", "-", "Chamarras","-", "-", "Pantalones", "-"]
ax.set_xticks(range(9))
ax.set_xticklabels(names)
plt.show()
```

4. Planteamos gráficos donde comparemos los contrastes de precios y promedios.





En conclusión las herramientas que vimos a lo largo del curso nos sirvieron para la creación de un web scrapper que en un dado caso lo podríamos vender para que las personas en un dado caso

puedan encontrar promociones de sus tiendas favoritas de ropa, marcas populares como Zara, Guess o entre otras

Por otra parte mostrar gráficos o la utilización de consultas en sql, para un mayor aprovechamiento y ahorro de tiempo.