Universidad Nacional Autónoma de México

Facultad de Ciencias

Manejo de Datos, Grupo 9157.

Proyecto Final Web Scraper

Imparten: Jessica Santizo Galicia

Ayudante: Sergio Alejandro Chávez Molotla

Integrantes

Angeles Aguillón Lily Avril.

Bustos Benítez Sonia Valeria.

Olguín Hérnandez Elisa Margarita.

¿QUÉ ES UN WEB SCRAPER?

Un web scraper es una herramienta que tiene como objetivo extraer información de sitios web, su principal ventaja es la automatización del proceso de búsqueda y selección de datos. Actualmente los web scraper son muy útiles para las empresas, pues así pueden obtener datos de manera más eficiente y asignar a sus empleados otro tipo de tareas, además, permiten profundizar en temas específicos, recabando información más detallada para la planificación estratégica.

OBJETIVOS

El objetivo de este documento es explicar detalladamente el desarrollo de un web scraper enfocado en la extracción de información (específicamente de salas, recamaras y comedores) de 3 páginas web de muebles, las cuales son, Muebles Troncoso, Casa de las lomas Muebles y Muebles Pergo. Con el seguimiento de la información presentada, se espera que cualquier persona con conocimientos de Python pueda desarrollar su propio web scraper de sitios web de mueblerías.

PREPARACIÓN DE EQUIPO.

Antes de comenzar con el desarrollo del web scraper hay que tomar en cuenta que se debe tener instalado Anaconda, una vez hecho esto, abrir 'Anaconda prompt' y ejecutar el siguiente comando **pip install --user selenium==3.141.0** que nos permitirá instalar Selenium, el cual es una herramienta de código abierto para automatizar los navegadores web en muchas plataformas.
El siguiente paso es descargar **ChromeDriver**, para lo cual tenemos que conocer la versión que estamos utilizando de Chrome, una vez hecho esto, escribimos en el buscador 'ChromeDriver' y entramos en la primera liga, después descargamos la versión que tengamos y guardamos el archivo .exe.
Finalmente, en Anaconda Navigator, encontraremos **Jupyter Notebook**, el cual será la herramienta que utilizaremos para el desarrollo del proyecto, una vez iniciado un archivo nuevo, importamos las siguientes paqueterías.

```
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
```

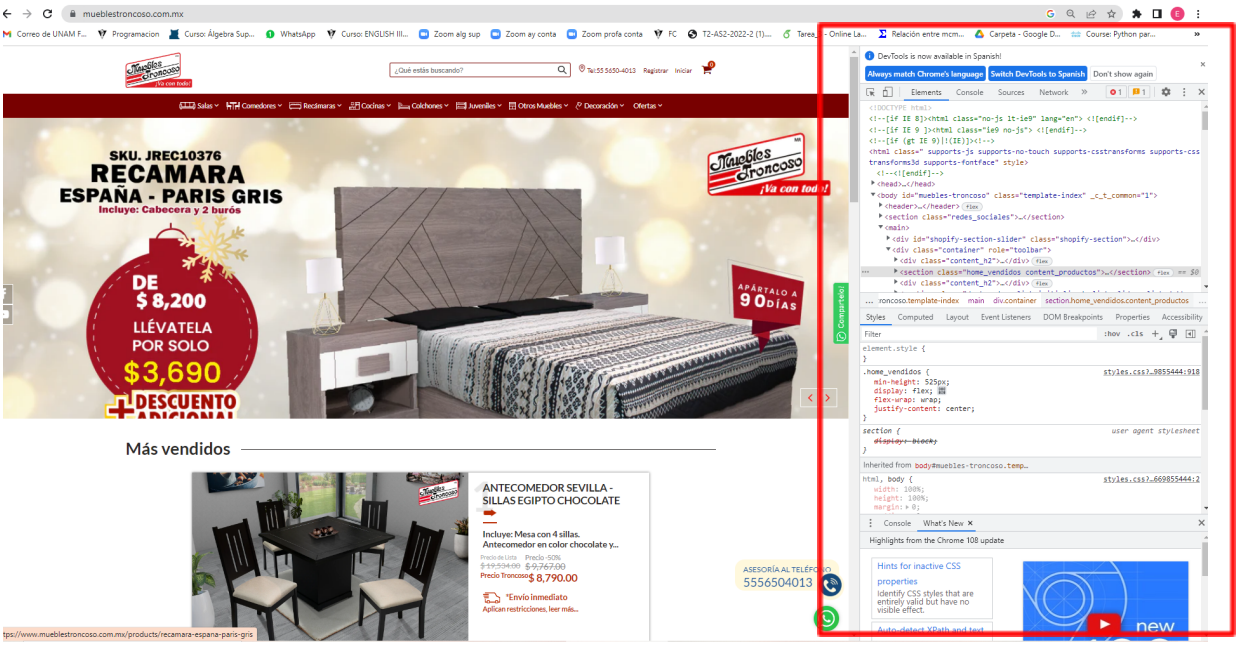
DEFINIR INFORMACIÓN A EXTRAER

De las 3 mueblerías nos centraremos en 3 productos, comedores, salas y recámaras, de los cuales la información a extraer es la fecha en que se hizo la consulta, el nombre de la muebleria, la marca, la descripción, el precio original y precio de promoción de cada uno de los productos.

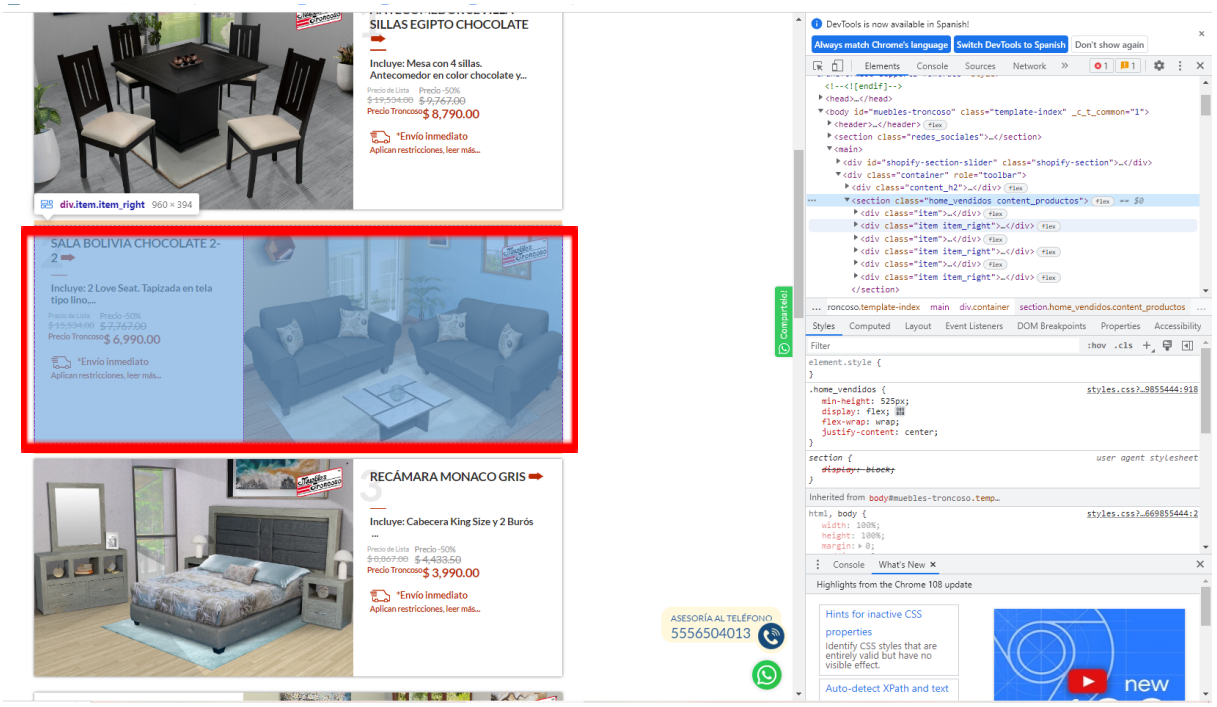
BÚSQUEDA DE CLASES EN SITIOS WEB

Es necesario buscar la clase que contiene todos los productos mostrados en cada página, para lo cual daremos el ejemplo de cómo buscarlo en la página de Muebles Troncoso, ya que la búsqueda de esta clase se hace de manera análoga en las otras 2 páginas.

El primer paso es ir al sitio web ‘Muebles Troncoso’, una vez dentro, damos click derecho en cualquier parte de la página y presionamos en inspeccionar, se desplegará una pestaña del lado derecho de la ventana y aparecerá el código en el que está desarrollado el sitio.



Posteriormente vamos a buscar la clase que contiene los productos, podemos ir pasando el cursor por el nombre de las clases y cuando se ilumine en azul el recuadro de cada producto, estamos en su clase, el nombre de esta, será el mismo para todos los productos del sitio, en el caso de Muebles Troncoso, la clase se llama **‘item’**.



En el caso de Casa de las lomas Muebles y Muebles Pergo, las clases se llaman **'item'** y **'sf__col-item'** respectivamente.

SOLICITUD AL SERVIDOR

Para continuar con el desarrollo del web scraper, lo siguiente es definir una función que es análoga para los tres sitios web, la cual se encargará de generar un data frame con la información que definimos por extraer, el parámetro de la función es el producto del que queremos los datos, por ejemplo si queremos de salas de Casa de las Lomas Muebles, obtendremos un data frame con la información de las salas que aparecen en la primera página del sitio web.
Mostraremos cómo se hace la función para el sitio de Casa de las Lomas Muebles, ya que para las otras dos mueblerías se haría de manera muy similar, solamente cambiando datos particulares.
Comenzamos definiendo el nombre de la función ***scraper_lomas(marca)***, después hacemos la carga del web driver en una variable llamada path, para realizarlo, asignamos la ruta donde tenemos el archivo .exe del Chrome Driver que descargamos, en una variable llamada driver, asignamos la función webdriver.Chrome con el parametro path, por último, para finalizar con esta parte para iniciar el navegador de Google Chrome, ponemos la función driver.get con parámetro la url de la página con un espacio para la marca, que se completará con el nombre del producto que estemos buscando, es decir **driver.get('https://www.casadelaslomas.com/'+marca+'.html').** Después de este proceso se abrirá una pestaña de Chrome con la dirección a la que nos estamos

dirigiendo, hasta este momento usamos la función sleep para pausar la ejecución de la función.
Una vez hecho lo anterior, usaremos el nombre de la clase de los productos que anteriormente ya habíamos buscado, vamos a definir una variable **productos**, igualandola a la función **driver.find_elements_by_class_name('item'),** notemos que aquí es donde estamos usando el nombre de la clase como parámetro, esto nos permitirá obtener una lista con todos los productos (que hayamos escogido) que aparezcan en la primera página. Nuevamente usamos la función sleep.
Posteriormente creamos listas vacías que contendrán los datos que queremos obtener, por ejemplo la lista_fecha, lista_autoservicio, lista_precios y así con cada parámetro de información por obtener. A continuación vamos a crear un ciclo for que nos ayudará para llenar las listas que definimos con la información que le corresponde a cada una y que está contenida en una sola lista que llamamos productos, por lo que estaremos recorriendo cada índice que se encuentre entre el rango 0 y la longitud de la lista de productos. A la lista fecha, con el método append le agregamos la fecha, que le podemos definir con la función time.strftime, para la lista de marca, mediante if y else if, dependiendo de si la marca es una sala, comedor o recamara, la agregamos a esta lista, para la lista nombre del producto, precio y precio de oferta, vamos a buscar la clase que contiene estos datos como lo hicimos para las otras clases y vamos a hacer uso de un try usando lo siguiente, por ejemplo el nombre de la clase que tiene el nombre del producto se llama "product name" entonces vamos a agregar los datos de la siguiente manera **lista_nombres_lomas.append(productos[i].find_elements_by_class_name("product-name")[0].text),** y procedemos de la misma forma para las otras dos listas, haremos uso de un except en el caso de que no haya datos, dentro del except para el ejemplo que estamos haciendo, sería **lista_nombres_lomas.append(np.nan),** en general esta parte de la función quedaría de la siguiente manera.

```
try:
    lista_nombres_lomas.append(productos[i].find_elements_by_class_name("product-name")[0].text)
    lista_preciosnormales_lomas.append(productos[i].find_elements_by_class_name("old-price")[0].text)
    lista_preciosespeciales_lomas.append(productos[i].find_elements_by_class_name("special-price")[0].text)
except:
    lista_nombres_lomas.append(np.nan)
    lista_preciosnormales_lomas.append(np.nan)
    lista_preciosespaciales_lomas.append(np.nan)
```

Después vamos a volver a dormir el equipo con la función sleep, y a continuación vamos a crear el data frame, le llamaremos df_lomas, tendrá las siguientes columnas: fecha, autoservicio, producto, nombre, precio normal y precio de promoción, cada una de estas columnas las vamos a igualar a las listas que obtuvimos anteriormente.
El siguiente paso es limpiar los datos, especialmente los precios, borramos el signo $, y como los dos precios estaban en una misma clase, con la función split convertimos el texto a una lista y los separamos en dos, es decir:

```
df_lomas[['textadicional','PRECIO_NORMAL']] = df_lomas.PRECIO_NORMAL.str.split("$",expand=True)
df_lomas[['textadicional2','PRECIO_PROMOCION']] = df_lomas.PRECIO_PROMOCION.str.split("$",expand=True)
```

Podemos convertir a mayúsculas los nombres con la función upper y a los precios quitarle las comas con replace y convertirlos a float con la función astype.

Por último, por medio de un return, devolvemos df_lomas, que será el data frame que contendrá todos los datos ordenados y limpios de salas, comedores o recamaras (según solicitamos).

Debemos tener en cuenta que, hasta ahora tenemos lista la función de la mueblería Casa de las Lomas, por lo que hará falta terminar las otras dos funciones, las cuales serán idénticas, a excepción de los nombres de las clases utilizadas y la dirección de cada sitio.

Ahora si, teniendo las tres funciones, podemos definir variables que serán el data frame de salas, comedores y recamaras de cada mueblería, lo podemos realizar de la siguiente forma:

```
# Salas
salas_lomas = scrapper_lomas('salas')
salas_troncoso = scrapper_troncoso('salas')
salas_pergo = scrapper_pergo('salas')

# Comedores
comedores_lomas= scrapper_lomas('comedores')
comedores_troncoso = scrapper_troncoso('comedores')
comedores_pergo= scrapper_pergo('comedores')

# Recamaras
recamaras_lomas = scrapper_lomas('recamaras')
recamaras_troncoso = scrapper_troncoso('recamaras')
recamaras_pergo = scrapper_pergo('recamaras')
```

Para concatenar las data frame de las 3 mueblerias y los 3 productos, usamos la función pd.concat y como parámetro las variables que acabamos de definir en el paso anterior, haciendo esto, obtendremos el siguiente data frame en general:

	FECHA	AUTOSERVICIO	PRODUCTO	NOMBRE	PRECIO_NORMAL	PRECIO_PROMOCION
0	12/12/2022	CASA DE LAS LOMAS	SALAS	SALA MODULAR DERECHA QUINCY GRAPHITE	28248.75	22599.0
1	12/12/2022	CASA DE LAS LOMAS	SALAS	SALA MODULAR IZQUIERDA EBONY KLEIN SMOKE	31248.75	24999.0
2	12/12/2022	CASA DE LAS LOMAS	SALAS	SALA MODULAR IZQUIERDA DARIA VL STONE GREY	33748.75	26999.0
3	12/12/2022	CASA DE LAS LOMAS	SALAS	SOFÁ BEIGE 1019	25873.75	20699.0
4	12/12/2022	CASA DE LAS LOMAS	SALAS	LOVE SEAT BEIGE 1019	19373.75	15499.0
...	...	...	...	...	...	...
431	12/12/2022	MUEBLES PERGO	RECÁMARAS	SET CAMA Y 2 BUROS TERUEL	104000.00	72990.0
432	12/12/2022	MUEBLES PERGO	RECÁMARAS	SET CAMA + BURÓS DINAMARCA	186000.00	79990.0
433	12/12/2022	MUEBLES PERGO	RECÁMARAS	SET CAMA Y 2 BUROS MCCOBB	104000.00	72990.0
434	12/12/2022	MUEBLES PERGO	RECÁMARAS	SET CAMA MÁS DOS BUROS MILAN SLATTE	126500.00	62990.0
435	12/12/2022	MUEBLES PERGO	RECÁMARAS	SET CAMA + 2 BURÓS SOHO II	116100.00	80990.0

436 rows × 6 columns

EXTRACCIÓN A BASE DE DATOS

FInalmente podemos llevar esta tabla que contiene toda la información recabada a un archivo de excel, con la función df_mueblerias.to_excel usando como parámetro el nombre del archivo de excel que vamos a crear, obtendremos en la carpeta donde estamos guardando el archivo de Jupyter nuestro archivo nuevo.

```
df_mueblerias.to_excel("scraper_mueblerias.xlsx")
```

CONCLUSIONES

El realizar este proyecto nos pareció muy interesante e importante para conocer las aplicaciones que podemos darle a la programación en la vida cotidiana ya que el desarrollo de un web scraper está involucrado en muchas empresas hoy en día debido a que los procesos de encontrar y recabar información se automatizan y con ello conseguimos disminuir la carga de trabajo, aumentar la velocidad de procesos y eliminar el error humano.

El conocer cómo obtener información de sitios web, en nuestro caso de 3 mueblerías nos parece que puede tener un gran potencial, pues hoy en día la información es una herramienta sumamente valiosa para optimizar diferentes procesos y tomar decisiones más acertadas, como equipo vemos posibilidad de que en un futuro el desarrollo de este proyecto en un ámbito laboral puede tener amplias posibilidades de apoyar a una empresa a visualizar datos de una forma más amena y fácil de interpretar.