Universidad Nacional Autónoma de México

Facultad de ciencias

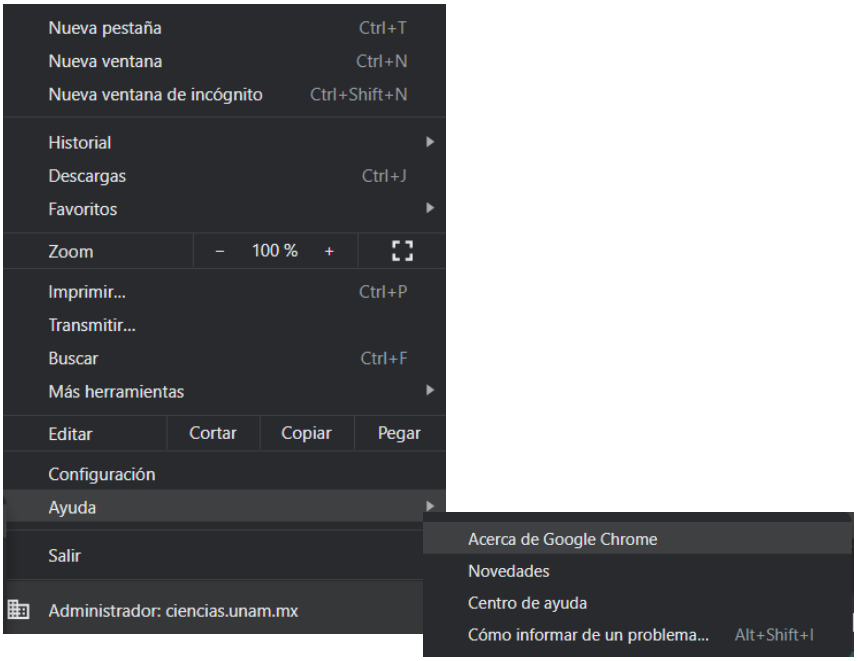
Manejo de datos

Webscraper

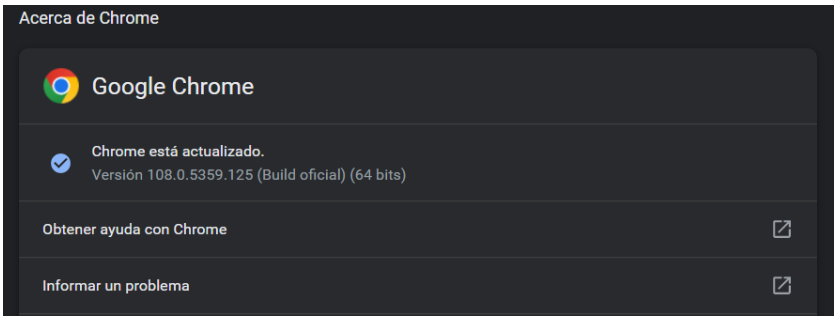
Integrantes:
Rodríguez Moreno Marcos Isaac
Contreras Miguel José Manuel
Del Rosario Jácome Rodrigo

Prof:
Jessica Santizo Galicia
Sergio Alejandro Chávez Molotla

Antes de iniciar a programar, tenemos que tener instaladas ciertas cosas.
Una de las principales es el “webdriver” el cual nos ayudará a poder acceder a una página de Google Chrome. Para la correcta instalación, necesitamos ver qué versión de Chrome tenemos.



Dando clic en “Acerca de Google Chrome” podremos visualizar nuestra versión que tenemos.



Ahora si podemos instalar el webdriver, es recomendable guardarlo en lugar donde lo podamos encontrar fácilmente, ya que más adelante ocuparemos la ruta del archivo.
Tambien necesitaremos instalar “Anaconda Navigator”, para poder utilizar Jupyter

Un paso importante es instalar dentro de Anaconda Prompt, lo siguientes, escribiendo los siguientes comandos

- pip install --user selenium == 3.141.0
- pip install random_user_agent
- pip install fake_useragent
- pip install pandasql

Antes de mencionar cuáles y cómo importarlas, notemos que hay varias formas de poder realizar un *web scraper,* con "*selenium*" (el cual usaremos en esta ocasión, ya que no es tan complicado de emplear y brinda seguridad para los posibles baneos), "*beautifulsoup", "requests".*

Ahora si podemos comenzar a programar, lo primero que tenemos que hacer es importar lo siguiente;

```
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.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
from datetime import date
from datetime import datetime
import matplotlib.pyplot as plt
```

Después, definamos qué es lo que queremos hacer, ¿Qué productos estamos buscando?, ¿De qué tienda ?, ¿Qué información queremos?

En este caso lo que queremos son productos electrónicos (Celulares, Consolas, Audífonos, Relojes). Las tiendas que buscaremos serán, Walmart, Palacio de Hierro y Soriana.

Recomendamos tener una o dos páginas extra, en ocasiones unas funcionan mejor que otras, de igual manera tener uno o dos productos extra en caso de no encontrarlos en todas las tiendas.

Queremos buscar y crear un DataFrame con los siguientes datos:

- Nombre del producto
- Marca
- Precio original
- Precio con descuento (en caso de existir)
- Autoservicio
- Fecha (de recolección de los datos)

Lo primero que tenemos que hacer es definir la función y dentro de ella poner la palabra *producto*, luego creamos una variable para cargar el webdriver.

```
def scrapper_palacio(producto):

        path = "C:\\webdriver\\chromedriver.exe"
        driver = webdriver.Chrome(path)
```

Aquí, pondremos la ruta donde se encuentra nuestro chromedriver

Ocuparemos una función en repetidas ocasiones que es “time.sleep()” la cual nos sirve para dormir la pagina por unos segundos, los que indiquemos nosotros, para que de tiempo de abrir la pagina, como esta ocupara internet no siempre es muy veloz.

El siguiente paso es, declarar la variable “url”, donde pegaremos la url de la página con la que trabajaremos y haremos unas pequeñas modificaciones

```
url = "https://www.elpalaciodehierro.com/buscar?q="+producto+"."
driver.get(url)
time.sleep(10)

items = driver.find_elements_by_class_name("b-product")
```

Ahora, tenemos que poner la palabra que pusimos al momento de definir la función en este caso es *producto,* pero para esto al momento de copiar la url de la página tendremos que fijarnos en donde cambia al momento de buscar un producto.

El siguiente paso es definir la variable items para poder utilizar la función, “driver.find_element_by_class_name” la cual nos ayudará a extraer toda la información del producto en este caso es “b-product”

Este nos servirá para saber cuántos productos hay la primera hoja de la página,
Tenemos que buscar ahora la marca, se hace de la misma manera, pero buscando en el html la parte donde subraya la marca, haremos un ciclo for para que lo haga en cada uno de los productos de la primera hoja.
También es necesario hacer una excepción en caso de que no exista y en la tabla lo llene con un NaN.

```
lista_marcas = []
for item in items:
    try:
        lista_marcas.append(item.find_element_by_class_name("b-product_tile-brand").text)
    except:
        lista_marcas.append(np.nan)
```

Esto lo repetiremos, para cada una de las cosas que buscamos, para el nombre del producto, el precio y precio promoción.

Teniendo todo esto, procedemos a crear el Data Frame, en la columnas pondremos los siguientes datos;
FECHA, AUTOSERVICIO, MARCA, NOMBRE, PRECIO, PROMOCION

La fecha la podemos poner utilizando “today = date.today()” y el autoservicio, será de la tienda donde estamos recolectando la información.

Este procedimiento lo repetimos para las dos tiendas restantes.
Una parte muy importante es generar las gráficas, para la fácil comprensión de los datos, es necesario importar lo siguiente:

```
import matplotlib.pyplot as plt
```

Para generar las gráficas tenemos que tener en claro los datos que queremos extraer, y que es lo que queremos resaltar de la recopilación de datos.