```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

### Paqueterias que se utilizaran

import pandas as pd
import pandasql as ps
import time
from selenium import webdriver
import matplotlib.pyplot as plt
import re
import numpy as np


### Creamos un DataFrame vacio y luego almacenamos los datos del
DataFrame
### en un archivo excel

aux=pd.DataFrame()
aux.to_excel("df_proyectofinal.xlsx",index=False)

"""
Creamos nuestra primera función la cual se encargara de recopilar los
datos de nuestros
productos en un DataFrame, en esta primera funcion especificamente
recopilaremos la informacion
del sitio web de Palacio de Hierro
"""

def Buscador_Precios_Selenium_PalaciodeHierro(producto):
    ### ingresamos a la pagina web
    path ="/usr/local/bin/chromedriver"
    driver=webdriver.Chrome(path)
    #Obtenemos el dominio de la pagina que nos permitira buscar los
productos por nombre
    url= "https://www.elpalaciodehierro.com/buscar?q="+producto
    #Accedemos al url en el navegador
    driver.get(url)

    ### Accedemos a la clase donde se realiza la descripcion general de
nuestro
    ### producto

    productos= driver.find_elements_by_class_name("b-product")

    ### accedemos a las urls almacenadas en la variable productos a
traves de una etiqueta
    ### en dado caso que no se encuentre el url se rellenara el espacio
con un nan,
    ### almacenamos todo lo anterior en una nueva lista.

    lista_urls=list()
    for i in range(len(productos)):
        try:

lista_urls.append(productos[i].find_element_by_tag_name("a").get_attrib
```

```
ute("href"))
        except:
            lista_urls.append(np.nan)


    ### Accedemos a los nombres de los productos a través de una
etiqueta de html
    ### y los almacenamos en una lista


    lista_nombres=list()
    for i in range(len(productos)):
        try:

lista_nombres.append(productos[i].find_elements_by_tag_name("a")
[1].text)
        except:
            lista_nombres.append(np.nan)

    ### accedemos a los precios base y promo de los productos a través
de las clases
    ### encontradas en los sitios web y almacenamos tanto los precios
base como los precios
    ### promo en 2 listas diferentes

    lista_precios=list()
    lista_promos=list()
    for i in range(len(productos)):
        try:

lista_precios.append(productos[i].find_elements_by_class_name("b-
product_price-value")[0].text)
        except:
            lista_precios.append(np.nan)
        try:

lista_promos.append(productos[i].find_elements_by_class_name("b-
product_price-value")[1].text)
        except:
            lista_promos.append(np.nan)

    ### Creamos un DataFrame a partir de las listas creadas con
anterioridad
    ### y anexamos a el "autoservicio", "marca" y la fecha de consulta.

    df_PH
=pd.DataFrame({"nombre":lista_nombres,"url":lista_urls,"precio1":lista_
promos,"precio2":lista_precios})
    df_PH["autoservicio"]="palacio de hierro"
    df_PH["marca"]= producto
    df_PH["fecha"]= time.strftime("%d/%m/%y")

    #Reordenamos las columnas de nuestro DataFrame

    df_PH =
df_PH[["fecha","autoservicio","marca","nombre","url","precio1","precio2
"]]
```

```
    df_PH  =df_PH.reset_index(drop=True)

    datos_webscraper=pd.read_excel("df_proyectofinal.xlsx")

    ###Se concatena nuestro archivo excel con nuestro DataFrame
    ###que contiene la informacion de los productos

    datos_webscraper= pd.concat([datos_webscraper,df_PH],axis=0)

    datos_webscraper.to_excel("df_proyectofinal.xlsx",index=False)

    ###Se cierran sucesivamente todas las paginas del consultadas en el
navegador y
    ### se termina la sesion webdriver

    driver.quit()
    return df_PH

def Buscador_Precios_Selenium_Sanborns(producto):
    ### ingresamos a la pagina web
    path ="/usr/local/bin/chromedriver"
    driver=webdriver.Chrome(path)
    url= "https://www.sanborns.com.mx/resultados/q="+producto+"/1"
    driver.get(url)

    ### Accedemos a los elementos que contienen los datos que queremos
de
    ### la pagina web

    productos= driver.find_elements_by_class_name("cardProduct")

    ### accedemos a las urls almacenadas en la variable productos

    lista_urls=list()
    for i in range(len(productos)):
        try:

lista_urls.append(productos[i].find_element_by_tag_name("a").get_attrib
ute("href"))
        except:
            lista_urls.append(np.nan)

    ### accedemos a los nombres de los productos

    lista_nombres=list()
    for i in range(len(productos)):
        try:

lista_nombres.append(productos[i].find_elements_by_tag_name("a")
[2].text)
        except:
            lista_nombres.append(np.nan)

    ### accedemos a los precios base y promo de los productos

    lista_precios=list()
```

```
    lista_promos=list()
    for i in range(len(productos)):
        try:

lista_precios.append(productos[i].find_elements_by_class_name("infoDesc
")[0].text.split()[0])
        except:
            lista_precios.append(np.nan)
        try:

lista_promos.append(productos[i].find_elements_by_class_name("infoDesc"
)[0].text.split("\n")[1])
        except:
            lista_promos.append(np.nan)

    df_sanborns
=pd.DataFrame({"nombre":lista_nombres,"url":lista_urls,"precio1":lista_
promos,"precio2":lista_precios})
    df_sanborns["autoservicio"]="sanborns"
    df_sanborns["marca"]= producto
    df_sanborns["fecha"]= time.strftime("%d/%m/%y")

    df_sanborns =
df_sanborns[["fecha","autoservicio","marca","nombre","url","precio1","p
recio2"]]

    df_sanborns  =df_sanborns.reset_index(drop=True)

    datos_webscraper=pd.read_excel("df_proyectofinal.xlsx")

    ### Se concatena nuestro archivo excel con nuestro DataFrame
    ### que contiene la informacion de los productos
    datos_webscraper= pd.concat([datos_webscraper,df_sanborns],axis=0)

    datos_webscraper.to_excel("df_proyectofinal.xlsx",index=False)

    driver.quit()
    return df_sanborns

def Buscador_Precios_Selenium_BA(producto):
    ### ingresamos a la pagina web
    path ="/usr/local/bin/chromedriver"
    #path= mipath
    driver=webdriver.Chrome(path)
    url= "https://www.bodegaaurrera.com.mx/productos?Ntt="+producto
    driver.get(url)

    ### Accedemos a los elementos que contienen los datos que queremos
    ### de la pagina web

    productos=
driver.find_elements_by_class_name("grid_product__30OQa")

    ### Accedemos a las urls almacenadas en la variable productos

    lista_urls=list()
    for i in range(len(productos)):
```

```
        try:

lista_urls.append(productos[i].find_element_by_tag_name("a").get_attrib
ute("href"))
        except:
            lista_urls.append(np.nan)

    ### Accedemos a los nombres de los productos

    lista_nombres=list()
    for i in range(len(productos)):
        try:

lista_nombres.append(productos[i].find_elements_by_tag_name("a")
[1].text)
        except:
            lista_nombres.append(np.nan)

    lista_precios=list()
    lista_promos=list()
    for i in range(len(productos)):
        try:

lista_precios.append(productos[i].find_elements_by_class_name("product_
price__2NBjj")[0].text.split("\n")[0])
        except:
            lista_precios.append(np.nan)
        try:

lista_promos.append(productos[i].find_elements_by_class_name("product_p
rice__2NBjj")[0].text.split("\n")[1])
        except:
            lista_promos.append(np.nan)


    df_BA
=pd.DataFrame({"nombre":lista_nombres,"url":lista_urls,"precio1":lista_
promos,"precio2":lista_precios})
    df_BA["autoservicio"]="bodega aurrera"
    df_BA["marca"]= producto
    df_BA["fecha"]= time.strftime("%d/%m/%y")

    df_BA =
df_BA[["fecha","autoservicio","marca","nombre","url","precio1","precio2
"]]

    df_BA  =df_BA.reset_index(drop=True)

    datos_webscraper=pd.read_excel("df_proyectofinal.xlsx")

    ### Se concatena nuestro archivo excel con nuestro DataFrame
    ### que contiene la informacion de los productos

    datos_webscraper= pd.concat([datos_webscraper,df_BA],axis=0)

    datos_webscraper.to_excel("df_proyectofinal.xlsx",index=False)
```

```
    driver.quit()
    return df_BA


### Creamos un ciclo for que itera sobre cada producto y a partir de
las funciones
### almacena  en un dataframe todos los datos adquiridos de los
distintos producto
### en cada sitio

for productos in ["iphone","beats","laptop"]:
    Buscador_Precios_Selenium_PalaciodeHierro(productos)
    Buscador_Precios_Selenium_Sanborns(productos)
    Buscador_Precios_Selenium_BA(productos)

df_PH=pd.read_excel("df_proyectofinal.xlsx")
df_PH

df_sanborns=pd.read_excel("df_proyectofinal.xlsx")
df_sanborns

df_BA=pd.read_excel("df_proyectofinal.xlsx")
df_BA

"""
Esta funcion lo que hace es editar las columnas de precio1 y precio2
de tal forma que el formato del costo del producto no presente comas
ni tampoco el simbolo $ y crea un excel con este nuevo formato
"""
def precios_floats(datos):
    #### eliminamos el signo de pesos de ambas columnas

    for i in range(len(datos["precio1"])):
        try:

datos["precio1"].iloc[i]=datos["precio1"].iloc[i].strip("$")
        except:
            pass

    for i in range(len(datos["precio2"])):
        try:

datos["precio2"].iloc[i]=datos["precio2"].iloc[i].strip("$")
        except:
            pass


    ### quitamos la separacion de comas para miles

    datos["precio1"]=datos["precio1"].replace(",","",regex=True)
    datos["precio2"]=datos["precio2"].replace(",","",regex=True)

    ### convertimos los precios a numericos
    datos['precio1'] = pd.to_numeric(datos['precio1'], errors='coerce')
    datos['precio2'] = pd.to_numeric(datos['precio2'], errors='coerce')

    datos.to_excel("df_proyectofinal_limpio.xlsx",index=False)
```

```
    ### visualizamos los tipos de datos
    print(datos.dtypes)
    return datos


precios_floats(df_PH)
df_PH=pd.read_excel("df_proyectofinal_limpio.xlsx")
df_PH

precios_floats(df_sanborns)
df_sanborns=pd.read_excel("df_proyectofinal_limpio.xlsx")
df_sanborns

precios_floats(df_BA)
df_BA=pd.read_excel("df_proyectofinal_limpio.xlsx")
df_BA


###Accedemos a la ruta donde tenemos guardado nuestro excel

archivo = "/Users/Yahir/Desktop/df_proyectofinal_limpio.xlsx"
df_tiendas = pd.read_excel(archivo)

### Almacenamos en variables nuestras consultas para hacer graficas a
partir de las mismas

#Numero de productos de marca iphone que hay en cada tienda
consulta1 = ps.sqldf("select autoservicio,marca,count(marca) from
df_tiendas where (marca = 'iphone') group by autoservicio")
print(consulta1)


#Cantidad de productos de marca beats cuyo precio real es inferior a
1000
consulta2 = ps.sqldf("select autoservicio,marca,count(marca) from
df_tiendas where (marca = 'beats') and (precio2 < 1000) group by
autoservicio")
print(consulta2)

#Promedio del costo que tienen las laptops en cada tienda con respecto
a su costo real
consulta3 = ps.sqldf("select autoservicio,marca,avg(precio2) from
df_tiendas where (marca = 'laptop') group by autoservicio")
print(consulta3)

#Cantidad de productos con respecto a la marca beats que tienen
descuentos
consulta4 = ps.sqldf("select autoservicio,marca,count(marca) from
df_tiendas where (precio1 is not null) and (marca= 'beats')  group by
autoservicio")
print(consulta4)

#Cantidad de productos cuyo descuento es mayor a 2500 con respecto a la
marca laptop
consulta5 = ps.sqldf("select autoservicio,marca,count(marca) from
df_tiendas where  (precio1 not null) and (precio2-precio1 > 2500) and
```

```
(marca='laptop') group by autoservicio")
print(consulta5)

###GRAFICOS

#Grafico1
consulta1.plot(x="autoservicio",y="count(marca)",kind =
"bar",color="yellow")
plt.ylabel("Cantidad de productos")
plt.title("Cantidad de productos de la marca iphone que tiene cada
tienda")

#Grafico2
consulta2.plot(x="autoservicio",y="count(marca)", kind = "bar",
color="green")
plt.ylabel("Cantidad de Productos")
plt.title("Cantidad de productos de la marca beats cuyo precio sin
descuento es menor a 1000")

#Grafico3
consulta3.plot(x="autoservicio",y="avg(precio2)", kind = "bar",
color="blue")
plt.ylabel("Precio Promedio")
plt.title("Promedio del costo que tienen las laptops respecto a su
costo real")

#Grafico 4
consulta4.plot(x="autoservicio",y="count(marca)", kind = "bar",
color="grey")
plt.ylabel("Numero de productos que tienen descuento")
plt.title("Cantidad de productos de la marca beats que tienen algun
descuento")

#Grafico5
consulta5.plot(x="autoservicio",y="count(marca)", kind = "bar",
color="orange")
plt.ylabel("Cantidad de Productos")
plt.title("Cantidad de productos de la marca laptop cuyo descuento es
mayor a 2500")

plt.show()
```