

Importación de librerías

```
In [ ]: from selenium import webdriver # Webscrapping bot
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.common.exceptions import NoSuchElementException

from selenium.webdriver.firefox.options import Options as FirefoxOptions
from selenium.webdriver.chrome.options import Options as ChromiumOptions

from selenium.webdriver.common.by import By
```

```
In [ ]: import logging # Para generar logs
from logging.handlers import TimedRotatingFileHandler
from logging import Formatter

import datetime
import os
import time
```

```
In [ ]: import pandas as pd #Manejo de dataframes
```

Variables a modificar para adaptar el código

```
In [ ]: urlToScrap ="https://images.google.com/"
deleteOldLogs = True

pathToDF = "../Outputs/"
fileToDF = "ProdsToScrap.csv"

webdriverToUse = "chromium"
```

Pequeñas funciones de apoyo

```
In [ ]: # Esta función se ha creado para mejorar comprensión de código en la configuración

def UTCFormatter(logFormatter):
    """
    Recibe un formatter de logeo
    Devuelve el horario a tiempo GMT
    """
    logFormatter.converter = time.gmtime
    return logFormatter
```

Configuración de logs

```
In [ ]: # Se inicia el proceso de registro de logs a nivel de INFO.
        logger = logging.getLogger('ScrapLog')
        logger.setLevel(logging.INFO)

        # Variables que determinan apartados posteriores
        timestamp = datetime.datetime.utcnow().strftime('%Y%m%d_%H-%M-%S')
        filename=f'ScrapImages{timestamp}.log'
        formatter = logging.Formatter('[%(asctime)s] %(name)s {%(filename)s:%(lineno)s}')
```

```
In [ ]: '''
        Indican como se debe crear el archivo de log
        Si "deleteOldLogs" es True, sólo se conservará el último archivo de log
        '''

        if deleteOldLogs ==True:
            listFilesinCWD = os.listdir(os.getcwd())
            for element in listFilesinCWD:
                if element.endswith(".log"):
                    os.remove(os.path.join(os.getcwd(), element))

        fileHandler = logging.FileHandler(filename=filename)
        logging.Formatter.converter = time.gmtime

        fileHandler.setLevel(logging.INFO)
        fileHandler.setFormatter(UTCFormatter(formatter))
        logger.addHandler(fileHandler)
```

Importación de datos

Se importa el dataset especificado en las variables generales definidas anteriormente.

```
In [ ]: df = pd.read_csv(f"{pathToDF}{fileToDF}")
        # df.drop(columns=df.columns[0], axis=1, inplace=True)
```

Comprobación de tamaño y composición del dataset importado como dataframe de Pandas.

```
In [ ]: print(df.shape)
        df.head(2)
```

```
(19778, 3)
```

```
Out[ ]:   id  product_id          name
0    0         5645  Weleda Hombre Crema Hidratante
1    1         28743    Gynea Gestagyn Men
```

Lógica del Scrapping

La siguiente función obtiene la url del primer resultado que aparece en google imágenes con el nombre del producto que recibe.

```
In [ ]: def ScrapFunction(prodToScrap, urlToScrap, driver):  
    try:  
        logger.info(f"Started with: {prodToScrap}")  
  
        driver.implicitly_wait(5)  
        driver.delete_all_cookies()  
        driver.implicitly_wait(5)  
        driver.get(urlToScrap)  
        driver.implicitly_wait(5)  
  
        acceptCookie = driver.find_element(By.XPATH, "/html/body/div[2]/")  
        acceptCookie.click()  
    except:  
        pass  
  
    selectImageBox = driver.find_element(By.XPATH, "/html/body/div[1]/di")  
  
    selectImageBox.send_keys(prodToScrap)  
    selectImageBox.send_keys(Keys.ENTER)  
    driver.implicitly_wait(5)  
  
    selectImage = driver.find_element(By.XPATH, "/html/body/div[2]/c-wiz")  
    selectImage.click()  
  
    driver.implicitly_wait(5)  
    driver.refresh()  
    driver.implicitly_wait(5)  
  
    urlImage = driver.find_element(By.XPATH, "/html/body/div[2]/c-wiz/di")  
  
    logger.info(f"Scrapped: {urlImage}")  
  
    return urlImage  
  
except:  
    logger.info(f"FUNCTIONERROR: {prodToScrap}")  
    return None
```

Ejecución de Selenium. Tiene configurado tanto webdriver de *Chromium* como de *Firefox* por cuestiones de *debugging* (especialmente en términos de rendimiento). Finalmente se empleó geckodriver con *Firefox* (definido en variables generales anteriormente) en una instancia 'C2-standard-4' de *Google Cloud* con una duración total de 28 horas de cálculo.

El siguiente código traslada los nombres de los productos de uno en uno a la función anterior de Scrapeo y, con la url obtenida, lo graba en la columna generada "url" en el índice apropiado.

```

In [ ]: print("Starting Webscrapping!")

if webdriverToUse != "firefox":
    opts = ChromiumOptions()
    opts.add_argument("--no-sandbox")
    opts.add_argument("--incognito")
    opts.add_argument("start-maximized")
    opts.add_argument("window-size=1920,1080")
    opts.add_argument("--headless")
    opts.add_experimental_option("excludeSwitches", ["enable-automation"])
    opts.add_experimental_option('useAutomationExtension', False)
    driver = webdriver.Chrome(options=opts)
else:
    opts = FirefoxOptions()
    opts.add_argument("--no-sandbox")
    opts.add_argument("--incognito")
    opts.add_argument("start-maximized")
    opts.add_argument("window-size=1920,1080")
    opts.add_argument("--headless")
    driver = webdriver.Firefox(options=opts)

driver.set_page_load_timeout(30)
driver.set_window_size(1920, 1080)

for position, element in enumerate(df["name"].tolist()):
    try:
        urlScrapped = ScrapFunction(element,urlToScrap,driver)
        df.loc[df.index[position], 'url'] = urlScrapped
        if position % 10:
            df.to_csv("Scrap.csv")
    except:
        logger.info(f"Scrapped: {urlImage}")
        if position % 10:
            df.to_csv("Scrap.csv")
        continue

driver.close()

```

```

In [ ]: # Chequeo preliminar de resultados obtenidos
df.head()

```

```

Out[ ]:

```

	id	product_id	name	url
0	0	5645	Weleda Hombre Crema Hidratante	https://weledaint-prod.global.ssl.fastly.net/b...
1	1	28743	Gynea Gestagyn Men	https://www.gynea.com/wp-content/uploads/2018/...
2	2	68986	Endocare Tensage Ampollas	https://www.cantabrialabs.es/wp-content/upload...
3	3	9692	Lacer Colutorio Fluor+Xilitol Sabor Fresa	https://statics.promofarma.com/static/promofar...
4	4	81921	Age Protect Sérum Intensivo Multiacción Uriage	https://cdns3-2.primor.eu/62309-thickbox/age-p...

Exportación de resultados a formato CSV

```
In [ ]: # Importante no eliminar "index=False" para mantener homogeneidad de los da  
df.to_csv("ScrapDef.csv", index=False)
```