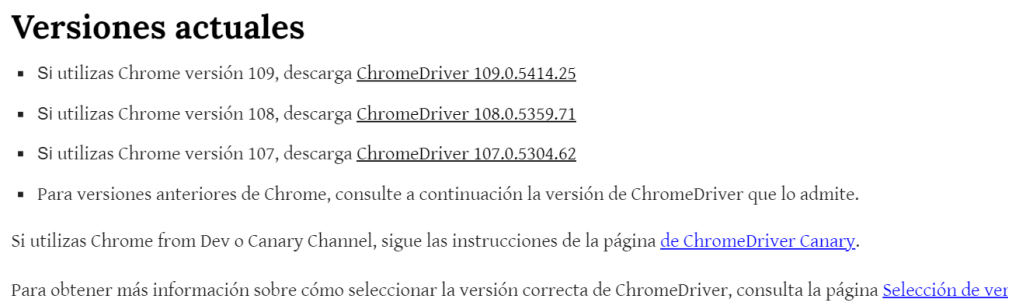
CÓMO CREAR UN WEB SCRAPER

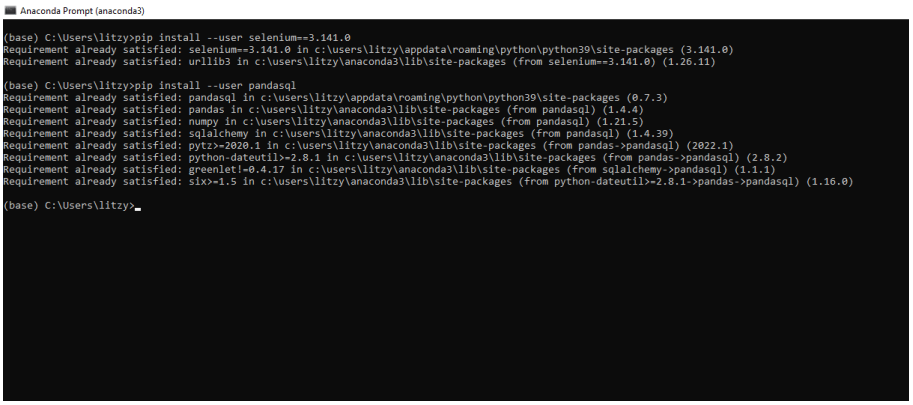
Empecemos por decir qué hace un web-scraper, bien, lo que hace un web-scraper es extraer información o datos de algunos sitios web, habrá sitios web donde no esté permitido el poder realizar un web-scraper. Ahora bien, los web-scraper son sumamente útiles hoy en día, es por eso que aquí podrás encontrar la información necesaria para que puedas realizar uno tú mismo.

1. Para lo anterior necesitarás tener instaladas un par de cosas, si no es así aquí te dejamos una liga en donde podrás descargarlas:
   - ❖ Anaconda: "Anaconda | Distribución de Anaconda"
   - ❖ Chrome: "Navegador web Google Chrome"
   - ❖ ChromeDriver: "ChromeDriver - WebDriver para Chrome - Descargas (chromium.org)". Antes de instalar ChromeDriver deberás verificar que versión tienes de Chrome, ¿Cómo puedo saber la versión de Google Chrome que estoy usando? (andro4all.com), y a partir de ellos descargar la versión más adecuada para tu máquina, a continuación te mostramos qué versiones hay y cómo se debe ver la página en la que debes descargar ChromeDriver.

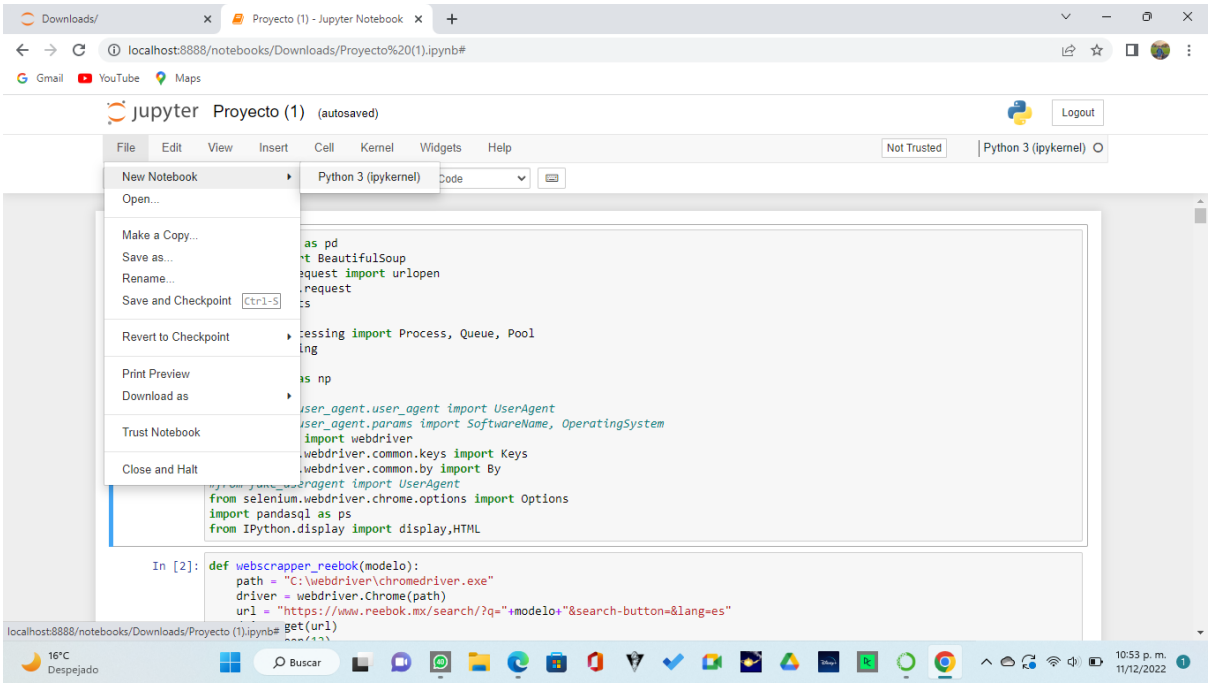


2. Cuanto tengas todo lo anterior deberás dirigirte a tu "Anaconda Prompt" y deberás instalar un par de cosas, para ello beras ingresar algunos comandos en "Anaconda Prompt":
   - ❖ Instalar selenium: pip install --user selenium==3.141.0
   - ❖ Instalar Pandas: pip install --user pandasql

   A continuación te mostramos cómo deberá verse tu terminal, si todo está en orden:



3. Crear una carpeta en los documentos de tu computadora.
4. Después de todo lo anterior, deberás abrir anaconda y posteriormente deberás abrir Júpiter, deberás seleccionar la carpeta que creaste y crear un documento como te lo mostramos a continuación:



5. Ya tienes un gran avance, para iniciar debes importar una serié de cosas en tu documento:

```
In [1]: 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
```

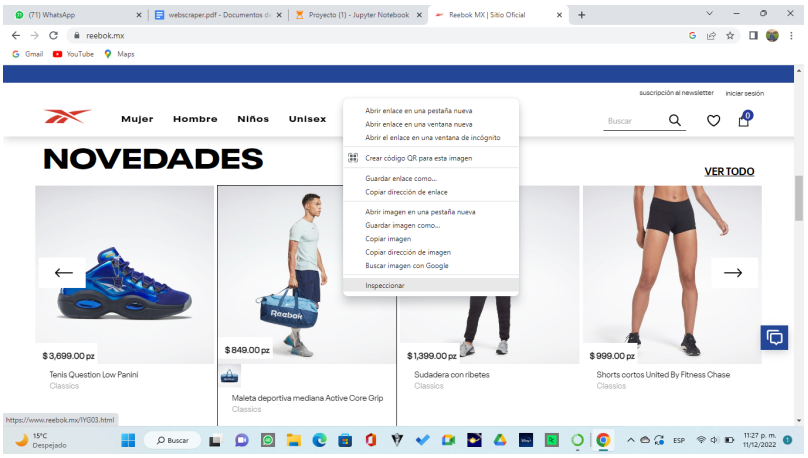
Para facilitar el proceso puedes copiar y pegar lo siguiente en tu documento:
“import pandas as pd
from bs4 import BeautifulSoup
from urllib.request import urlopen
import urllib.request
import requests
import time
from multiprocessing import Process, Queue, Pool
import threading

```
import sys
import numpy as np
import re
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
import pandasql as ps
from IPython.display import display,HTML”
```

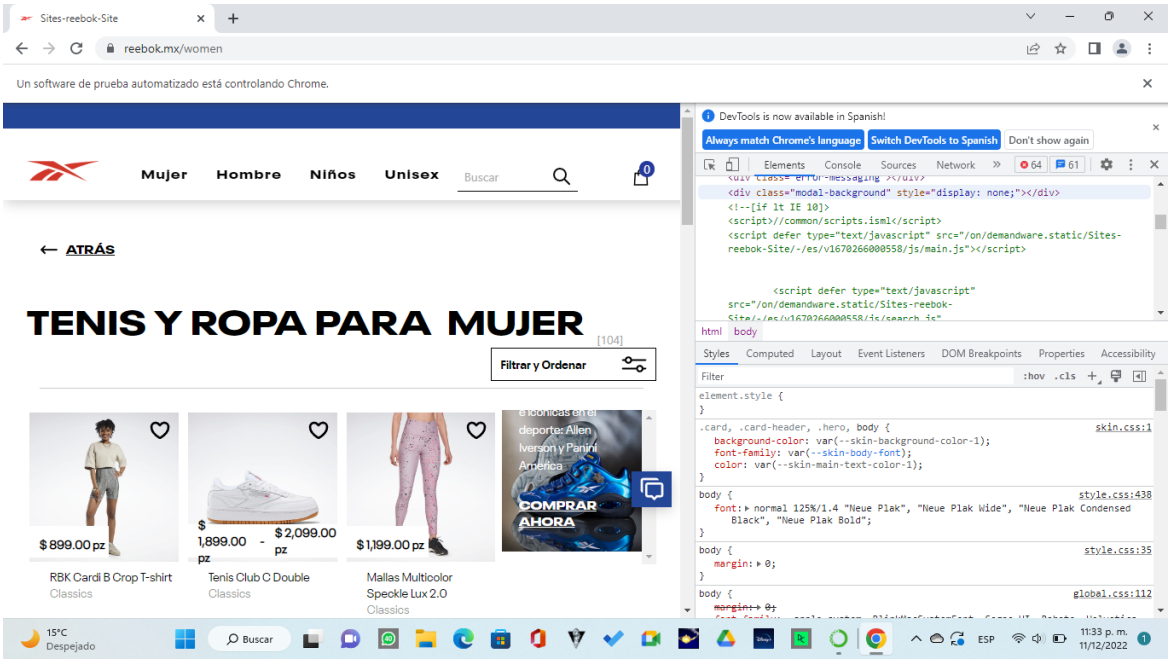
6. Elige una página web de donde quieras extraer datos e información de la misma.
7. Ahora empieza realmente el proceso del web-scraper, para esto debemos definir un par de cosas, por ejemplo:
   path = "C:\chromedriver\chromedriver.exe", donde esta es la url del chromeDriver que debiste descargar en el paso número 1.
   url="https://www.reebok.mx/search/?q="+modelo+"&search-button=&lang=es" donde esta es la url del sitió que elegiste con anterioridad

```
def webscrapper_reebok(modelo):
    path = "C:\chromedriver\chromedriver.exe"
    driver = webdriver.Chrome(path)
    url = "https://www.reebok.mx/search/?q="+modelo+"&search-button=&lang=es"
    driver.get(url)
    time.sleep(12)
```

8. Después viene una parte sumamente importante, el definir qué vamos a buscar, en este caso:
   productos=driver.find_elements_by_class_name("plp__grid-products-wrapper"), donde "plp__grid-products-wrapper" es la clase en donde encontramos los productos que queremos.
9. Es muy importante decir que el buscar la clase es el trabajo más importante de nuestro web scraper, para ello deberás ir a la página web que seleccionaste, al correr tu documento se abre automáticamente, y dar clic derecho sobre lo que desees encontrar información, en este caso seleccionaremos un producto, al dar clic derecho deberás ir a “inspeccionar”



Cuando estés en inspeccionar, debes buscar la clase que te de información de lo que has seleccionado, debes ser muy cuidadoso en este paso pues si seleccionas una clase que no es la correcta, no podrás obtener los datos que requieres.



Debes estar consciente de que este proceso se repite, pues cada “producto” está en una clase diferente, podrían llegar a coincidir, sin embargo, la probabilidad de que ocurra eso es casi nula.

A grandes rasgos, el “inspeccionar” es lo que nos facilita todo y donde literalmente encontramos todo para nuestro web scraper.

10. Para extraer información usaremos algunas funciones, que no son nada difíciles, incluso es posible que ya estés familiarizado con ellas, por ejemplo:
    - ❖ time.sleep()
    - ❖ len()
    - ❖ append()
    - ❖ entre otros
11. Una vez ubicada la clase donde se encuentra la información deseada, crearemos diferentes listas para asignar las diferentes cosas que deseemos, por ejemplo: la marca, los nombres del producto, los precios y sus descuentos.

```
lista_marcas = []
for i in range(0,len(productos)):
    lista_marcas.append("Reebok")

time.sleep(12)

lista_nombres = []
for i in range(0,len(productos)):
    try:
        productos[i].find_elements_by_class_name("link.product-tile__title")[0].text #Para el nombre
        lista_nombres.append(productos[i].find_elements_by_class_name("link.product-tile__title")[0].text)
    except:
        lista_nombres.append(np.nan)

time.sleep(12)

lista_precios = []
for i in range(0,len(productos)):
    try:
        productos[i].find_elements_by_class_name("value")[0].text #Para el nombre
        lista_precios.append(productos[i].find_elements_by_class_name("value")[0].text)
    except:
        lista_precios.append(np.nan)

time.sleep(12)

lista_categorias = []
for i in range(0,len(productos)):
    try:
        productos[i].find_elements_by_class_name("product-tile__category-wrapper")[0].text #Para el nombre
        lista_categorias.append(productos[i].find_elements_by_class_name("product-tile__category-wrapper")[0].text)
    except:
        lista_categorias.append(np.nan)
```

Para la creación de estas clases, podrás ocupar este código, sin embargo este servirá si le haces los cambios adecuados de acuerdo a lo que tu necesites en tu web -scraper.

12. Posteriormente, después de crear nuestras listas y modificarlas para obtener lo que buscamos, empezaremos con la creación del data frame, pero antes de crearlo, ¿qué es un data frame?
    Son estructuras de datos de dos dimensiones (rectangulares) que pueden contener datos de diferentes tipos, por lo tanto, son heterogéneas. Esta estructura de datos es la más usada para realizar análisis de datos.
    Cabe recordar que, este data frame, así como todos los pasos anteriores son en específico para la marca “Rebook”, pero este proceso es el mismo para cualquier tipo de producto donde se permita hacer un web-scraper.
    para la creación de este, ocuparemos:

```
df_reebok = pd.DataFrame(columns = ["fecha_consulta","marca","modelo","nombre","precios","categoría","colores"])
df_reebok["fecha_consulta"] = lista_fechas
df_reebok["marca"] = lista_marcas
df_reebok["modelo"] = lista_modelos
df_reebok["nombre"] = lista_nombres
df_reebok["precios"] = lista_precios
df_reebok["categoría"] = lista_categorias
df_reebok["colores"] = lista_colores

df_reebok.precios = df_reebok.precios.str.replace(",","")
df_reebok.precios = df_reebok.precios.str.replace("$","")
df_reebok.precios = df_reebok.precios.str.replace("pz","")
df_reebok.precios = df_reebok.precios.astype(float)

time.sleep(3)

driver.quit()
```

En la primera línea, asignamos nombres a las columnas y éstas se encontraran la fecha de consulta, marca, modelo, precio, categoría y el color del producto que deseamos buscar.
Después, en las líneas siguientes asignaremos el contenido de estas columnas, por ejemplo como se ve en la imagen anterior, en la segunda línea a la columna que recibe el nombre de marca, esta tendrá los elementos de la

lista que creamos anteriormente en el paso 11, de nombre “lista_marcas” y así sucesivamente, hasta terminar con todas las columnas.

13. Por motivos estéticos la columna de nombres la pondremos en mayúsculas.

```
#Ponemos la columna de nombres en mayúsculas
df_converse.NOMBRE = df_converse.NOMBRE.str.upper()
```

14. Debido a que en algunas páginas, no es tan sencillo obtener el precio o algún elemento que deseemos, entonces con ayuda de una expresión regular, en este caso ocupamos la función .replace, que nos ayudará a quitar las el signo de pesos ($) y las comas (,). de esta forma tendremos:

```
#Quitamos lo que no necesitamos para solo conseguir el precio, además de convertir a float el precio.
df_converse.PRECIO = df_converse.PRECIO.str.replace(",","")
df_converse.PRECIO = df_converse.PRECIO.str.replace("$","")
df_converse.PRECIO = df_converse.PRECIO.astype(float)

#df_converse = df_converse[df_converse.nombre.str.contains(modelo)]
```

15. Después hacemos una búsqueda para la creación del data frame, por ejemplo, buscamos el modelo “Chuck- Taylor“, además que en este paso se crea un documento Excel donde también ahí se encuentran las tablas con dicha información.

```
9]: #Buscamos el modelo 'Chuck-Taylor' y creamos el primer DataFrame
    Chuck_Taylor = webscrapper_converse("Chuck-Taylor")
    #Lo importamos a excel y lo nombramos así
    Chuck_Taylor.to_excel("Chuck_Taylor.xlsx")
    #Leemos el archivo que acabamos de crear
    Chuck_Taylor = pd.read_excel("Chuck_Taylor.xlsx",index_col= 0)
    #Chuck_Taylor

    C:\Users\Jorge Cortés\AppData\Local\Temp\ipykernel_58296\3991138106.py:88:
```

Y el resultado obtenido será este:

```
]:
```

	FECHA	AUTOSERVICIO	MODELO	NOMBRE	PRECIO	COLORES
0	15/12/2022	Converse	Chuck-Taylor	FUTURE METALS CHUCK TAYLOR ALT STAR EN BOTA DE...	1999.0	1 color
1	15/12/2022	Converse	Chuck-Taylor	FUTURE METALS CHUCK TAYLOR ALT STAR EN BOTA DE...	1999.0	1 color
2	15/12/2022	Converse	Chuck-Taylor	COUNTER CLIMATE CHUCK TAYLOR ALL STAR EN BOTA ...	2399.0	3 colores
3	15/12/2022	Converse	Chuck-Taylor	COUNTER CLIMATE CHUCK TAYLOR ALL STAR EN BOTA ...	2399.0	3 colores
4	15/12/2022	Converse	Chuck-Taylor	COUNTER CLIMATE CHUCK TAYLOR ALL STAR EN BOTA ...	2399.0	3 colores
...	...	...	...	...	...	...
104	15/12/2022	Converse	Ultra	CHUCK TAYLOR ULTRA EN MEDIA BOTA DE MATERIAL T...	2099.0	3 colores
105	15/12/2022	Converse	Ultra	CHUCK TAYLOR ALL STAR ULTRA EN MEDIA BOTA DE T...	1949.0	1 color
106	15/12/2022	Converse	Ultra	CHUCK TAYLOR ALL STAR ULTRA EN MEDIA BOTA DE T...	1949.0	1 color
107	15/12/2022	Converse	Ultra	CHUCK TAYLOR ALL STAR FLUX ULTRA EN MEDIA BOTA...	1899.0	1 color
108	15/12/2022	Converse	Ultra	FOUNDATION CHUCK TAYLOR ALL STAR FLUX ULTRA EN...	1899.0	1 color

109 rows × 6 columns

16. Una vez de hacer los pasos anteriores con todos los productos deseados, concatenamos todos en una sola tabla de esta forma:

```
#Ahora concatenamos los DataFrames de cada marca en uno solo
scrapper = pd.concat([scrapper_Reebok,scrapper_Converse,scrapper_Puma])
#Arreglamos el índice de las columnas
scrapper.index = range(scrapper.shape[0])
#Mandamos a llamar al DataFrame final
scrapper
```

Y el resultado sera este:

6]:

	FECHA	AUTOSERVICIO	MODELO	NOMBRE	PRECIO	COLORES
0	15/12/2022	Reebok	Royal	TENIS REEBOK ROYAL ULTRA	1599.0	NaN
1	15/12/2022	Reebok	Royal	TENIS ROYAL COMPLETE CLN 2	1099.0	NaN
2	15/12/2022	Reebok	Royal	TENIS ROYAL COMPLETE CLN 2	1099.0	NaN
3	15/12/2022	Reebok	Royal	TENIS REEBOK ROYAL PRIME 2	949.0	NaN
4	15/12/2022	Reebok	Royal	TENIS ROYAL BRIDGE 4 REEBOK	949.0	NaN
...	...	...	...	...	...	...
273	15/12/2022	Puma	Vogue	SUDADERA MUJER PUMA X VOGUE	1599.0	NaN
274	15/12/2022	Puma	Vogue	SUDADERA MUJER PUMA X VOGUE	1599.0	NaN
275	15/12/2022	Puma	Vogue	PANTS MUJER PUMA X VOGUE	1499.0	NaN
276	15/12/2022	Puma	Vogue	PANTS MUJER PUMA X VOGUE	1499.0	NaN
277	15/12/2022	Puma	Vogue	PANTS MUJER PUMA X VOGUE	1499.0	NaN

278 rows × 6 columns

17. Como se nos pide en el proyecto, haremos consultas en SQL, un ejemplo de sería este:
Aquí se busca tener todos los productos de mayor a menor precio:

]:
```
#Ordenando todos los productos de mayor a menor precio
ps.sqldf("select * from scrapper order by PRECIO desc")
```

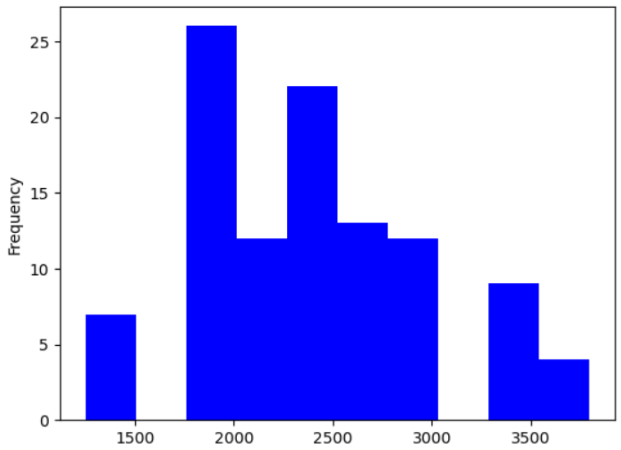
]:

	FECHA	AUTOSERVICIO	MODELO	NOMBRE	PRECIO	COLORES
0	15/12/2022	Converse	Choclo	ALL STAR BB SHIFT EN CHOCLO DE MATERIAL SINTÉTICO	3799.0	1 color
1	15/12/2022	Converse	Choclo	ALL STAR BB SHIFT EN CHOCLO DE MATERIAL SINTÉTICO	3799.0	1 color
2	15/12/2022	Converse	Choclo	VIVA LAS VEGAS ALL STAR BB SHIFT EN CHOCLO DE ...	3599.0	1 color
3	15/12/2022	Converse	Choclo	SEASONAL ALL STAR BB SHIFT EN CHOCLO DE MATERI...	3599.0	1 color
4	15/12/2022	Converse	Choclo	UTILITY EXPLORE COUNTER CLIMATE UTILITY EN CHO...	3449.0	2 colores
...	...	...	...	...	...	...
273	15/12/2022	Reebok	Playera	PLAYERA GRÁFICA TRAINING ESSENTIALS	399.0	None
274	15/12/2022	Converse	Chuck-70s	SEE BEYOND CHUCK 70 EN BOTA DE LONA	NaN	None
275	15/12/2022	Converse	Elevaciones	SEASONAL COLOR CTAS LUGGED 2.0 EN BOTA DE LONA	NaN	None
276	15/12/2022	Converse	Elevaciones	RUN STAR MOTION EN PLATAFORMA DE LONA	NaN	None
277	15/12/2022	Converse	Elevaciones	RUN STAR MOTION EN PLATAFORMA DE LONA	NaN	None

278 rows × 6 columns

18. Finalmente crearemos gráficos, de esta forma nos facilitará la interpretación de la información que acabamos de obtener.
    En esta parte puedes crear los gráficos de tu preferencia, en este caso para uno de nuestros gráficos ocupamos el siguiente:

```
[37]: #Comparando los precios de todos los Converse
Converse_hist = scrapper[scrapper.AUTOSERVICIO=="Converse"]["PRECIO"].plot(kind="hist",color="blue")
```



    Este gráfico nos permite comparar todos los precios de los productos de la marca Converse.

19. Un paso extra es ver que si corra tu web scraper y ser feliz viendo que tu creación si funciona mientras elevas tu ego.