Copilot: que es playwrite y cuales son sus alternativas

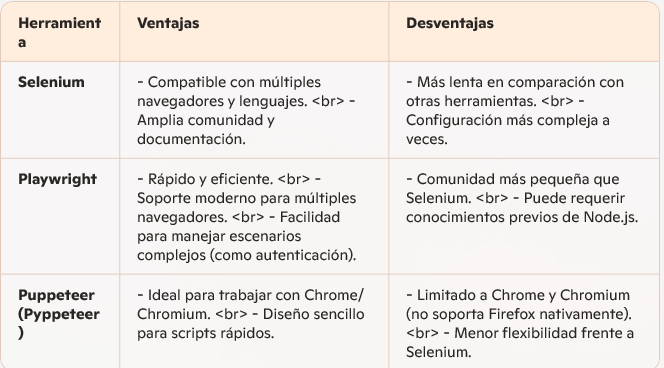
Playwright es una herramienta de código abierto desarrollada por Microsoft para realizar pruebas automáticas de aplicaciones web. Es especialmente útil para interactuar con navegadores modernos (como Chromium, Firefox y WebKit) y permite escribir pruebas que simulan acciones de los usuarios en la interfaz, como hacer clic, llenar formularios y navegar entre páginas. Es muy potente para asegurar la funcionalidad y la experiencia del usuario en aplicaciones web modernas.

Alternativas a Playwright:

Si necesitas más detalles sobre alguna de estas herramientas o una comparación más específica, puedo ayudarte a profundizar en ellas. ¿Qué opinas?

Copilot: de estas aplicaciones cuales son las mejores 3 para hacer una aplicación de webscraping con Python. presentar los resultados en una tabla comparativa

Para aplicaciones de web scraping con Python, las herramientas más recomendadas serían Selenium, Playwright y Puppeteer (a través de su implementación en Python llamada Pyppeteer). A continuación, te presento una tabla comparativa para que evalúes cuál se adapta mejor a tus necesidades:



Si necesitas más detalles sobre cómo comenzar con alguna de estas herramientas, puedo proporcionarte tutoriales o guías específicas. ¿Tienes alguna preferencia?

**Copilot:**

Escribir una aplicación Python utilizando selenium y otra utilizando playwright para obtener la cotización de Dólar U.S.A. para compra y venta identificando también la fecha de dicha cotización de la pagina https://www.bna.com.ar/

Aquí tienes ejemplos de cómo escribir aplicaciones Python para obtener la cotización del Dólar U.S.A. desde la página del Banco Nación usando Selenium y Playwright. Asegúrate de que estas librerías estén instaladas antes de ejecutar los scripts.

Con Selenium:

[BNAselenium.py](https://alumniiaeedu-my.sharepoint.com/:u:/g/personal/pszekieta_mail_austral_edu_ar/EWJOWlrM2SRPuV9NrC7bjGcBpjoJxx-L96N--ypqkIvprQ?e=xb4isQ)

Con Playwright

[BNAPlaywright.py](https://alumniiaeedu-my.sharepoint.com/:u:/g/personal/pszekieta_mail_austral_edu_ar/EU7O5gLhCyhNiiNX2nlp1iIBWvL5GjEcZY52H8Hf76YAvA?e=W06LKI)

Nota Importante sobre codigo:

Las etiquetas y clases de los elementos ("div.cotizacion\_compra, div.cotizacion\_venta , div.cotizacion\_fecha) son hipotéticas. Necesitas inspeccionar la página oficial del Banco Nación para verificar los selectores correctos usando las herramientas de desarrollo del navegador.

Si necesitas ayuda configurando tu entorno o aclarando algún paso, ¡avísame!

Copilot: que librerías necesito instalar para correr esta aplicación

**Selenium**

pip instal selenium

Instalar el driver del navegador:

Para Chrome, necesitas instalar ChromeDriver. Asegúrate de que la versión del driver sea compatible con tu navegador Chrome. Puedes descargarlo desde ChromeDriver.

**Playwright**

pip install playwright

playwright install

Otros requisitos comunes:

Python: Asegúrate de tener instalada una versión moderna (Python 3.7 o superior).

Herramientas adicionales: Podrías necesitar instalar pip si aún no lo tienes configurado en tu entorno

Copilot: dado el siguiente código modificarlo para que se conecte directamente desde colab a la pagina "[https://www.bna.com.ar/"](https://www.bna.com.ar/%22) , sin usar selenium y obtenga dolar\_compra = driver.find\_element(By.XPATH, '[//[@id="billetes"]/table/tbody/tr[1]/td[2]').text](mailto://[@id="billetes"]/table/tbody/tr[1]/td[2]').text) dolar\_venta = driver.find\_element(By.XPATH, '[//[@id="billetes"]/table/tbody/tr[1]/td[3]').text](mailto://[@id="billetes"]/table/tbody/tr[1]/td[3]').text)... el código a modificar es: from selenium import webdriver from selenium.webdriver.common.by import By import time

Prompt a DeepSeek para hacer un agente de recopilacion automatico:  
I am taking a web mining class to obtain the exchange rate of dollar to argentine peso from different websites. This is what I have so far: from selenium import webdriver from selenium.webdriver.common.by import By from selenium import webdriver from selenium.webdriver.chrome.options import Options import time # Configuración de Selenium para Chrome chrome\_options = Options() chrome\_options.add\_argument("--headless") driver = webdriver.Chrome(options=chrome\_options) try: # Abrir la página del BNA driver.get("https://www.bna.com.ar/") driver.maximize\_window() time.sleep(5) # Esperar a que la página cargue # Obtener la fecha de la cotización fecha\_cotizacion = driver.find\_element(By.CLASS\_NAME, 'fechaCot').text # Obtener la cotización del dólar (compra y venta) dolar\_compra = driver.find\_element(By.XPATH, '//\*[@id="billetes"]/table/tbody/tr[1]/td[2]').text dolar\_venta = driver.find\_element(By.XPATH, '//\*[@id="billetes"]/table/tbody/tr[1]/td[3]').text print(f"Fecha de la cotización: {fecha\_cotizacion}") print(f"Cotización del dólar (Compra): {dolar\_compra}") print(f"Cotización del dólar (Venta): {dolar\_venta}") finally: # Cerrar el navegador driver.quit() I also want to run an agent every day at certain time to obtain the exchange rate.

Y tambien: Great. Is there a way to leave the agent running but not the code open? Maybe in some part of the syste,