

Automatización en Amazon



Genera el código en Python utiliza Selenium para automatizar la navegación web y realizar una serie de acciones en la página de Amazon de México, espera a que el cuadro de búsqueda este presente y envía la búsqueda de "Discos duros SSD", espera a que aparezcan los resultados, selecciona solo aquellos de la marca "ADATA", espera a que haga la selección y ordénalos por menor precio y mantén el navegador abierto



Aquí tienes un ejemplo de código en Python utilizando Selenium para automatizar la navegación web en la página de Amazon de México y realizar las acciones solicitadas. Este script busca "Discos duros SSD", filtra por la marca "ADATA", y ordena los resultados por menor precio.

Primero, asegúrate de tener instaladas las siguientes bibliotecas:

- selenium
- webdriver-manager

Si no las tienes instaladas, puedes hacerlo usando pip:

```
sh
```

```
pip install selenium webdriver-manager
```

Luego, puedes usar el siguiente código:

```
python
```

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

# Configuración del controlador de Chrome
driver = webdriver.Chrome(ChromeDriverManager().install())

# Abre Amazon México
driver.get("https://www.amazon.com.mx")

# Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
search_box = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
)
search_box.send_keys("Discos duros SSD")
search_box.submit()

# Espera a que aparezcan los resultados
WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
)

# Filtra por la marca ADATA
```

```
filter_brand = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
)
filter_brand.click()

# Espera a que se realice la selección de la marca
WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
)

# Ordena por menor precio
sort_dropdown = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
)
sort_dropdown.click()

sort_low_to_high = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "s-result-sort-select_1"))
)
sort_low_to_high.click()

# Mantén el navegador abierto
print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")
```

Este código realiza las siguientes acciones:

1. Abre el navegador y navega a Amazon México.
2. Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda de "Discos duros SSD".
3. Espera a que aparezcan los resultados de la búsqueda.
4. Filtra los resultados por la marca "ADATA".
5. Espera a que se complete la selección de la marca.
6. Ordena los resultados por menor precio.
7. Mantiene el navegador abierto.

Asegúrate de tener Google Chrome instalado en tu sistema para que el controlador `ChromeDriver` funcione correctamente.



el código debe funcionar en chrome



A continuación te proporciono el código de Python utilizando Selenium y `webdriver-manager` para asegurar que funcione con Chrome. Este script automatiza la búsqueda de "Discos duros SSD" en Amazon México, filtra los resultados por la marca "ADATA" y los ordena por precio ascendente. Asegúrate de tener instaladas las bibliotecas necesarias:

```
sh
```

```
pip install selenium webdriver-manager
```

Aquí tienes el código:

python

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

# Configuración del controlador de Chrome
driver = webdriver.Chrome(ChromeDriverManager().install())

try:
    # Abre Amazon México
    driver.get("https://www.amazon.com.mx")

    # Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
    search_box.send_keys("Discos duros SSD")
    search_box.submit()

    # Espera a que aparezcan los resultados
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
    )

    # Filtra por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()

    # Espera a que se realice la selección de la marca
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
    )

    # Abre el menú de ordenamiento
    sort_dropdown = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
    )
    sort_dropdown.click()

    # Selecciona ordenar por menor precio
    sort_low_to_high = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "s-result-sort-select_1"))
    )
    sort_low_to_high.click()

    # Mantén el navegador abierto
    print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")

except Exception as e:
    print(f"Ocurrió un error: {e}")

finally:
    # No cerramos el navegador para que puedas verlo
    # driver.quit()
    pass
```

Este código hace lo siguiente:

1. Inicia el navegador Chrome y abre la página de Amazon México.
2. Espera a que el cuadro de búsqueda esté presente, luego realiza la búsqueda de "Discos duros SSD".
3. Espera a que los resultados de la búsqueda aparezcan.

4. Filtra los resultados para mostrar solo productos de la marca "ADATA".
5. Espera a que la selección de la marca se complete.
6. Abre el menú de ordenamiento y selecciona ordenar por precio ascendente.
7. Mantiene el navegador abierto para que puedas revisar los resultados.

Nota: El bloque `finally` contiene un comentario `# driver.quit()` que está deshabilitado para que el navegador no se cierre automáticamente. Puedes habilitarlo si deseas que el navegador se cierre al finalizar el script.



Mejora el código para que se mantenga abierto el navegador hasta que se pulse una tecla



Para mantener el navegador abierto hasta que se pulse una tecla, puedes utilizar la función `input()` de Python para pausar la ejecución del script hasta que el usuario presione una tecla. Aquí tienes el código mejorado:

python

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

# Configuración del controlador de Chrome
driver = webdriver.Chrome(ChromeDriverManager().install())

try:
    # Abre Amazon México
    driver.get("https://www.amazon.com.mx")

    # Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
    search_box.send_keys("Discos duros SSD")
    search_box.submit()


    # Espera a que aparezcan los resultados
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
    )

    # Filtra por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()

    # Espera a que se realice la selección de la marca
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
    )

    # Abre el menú de ordenamiento
    sort_dropdown = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
    )
    sort_dropdown.click()
```

1. **Instalación de dependencias:** El código utiliza las bibliotecas ``selenium`` y ``webdriver-manager``, asegúrate de tenerlas instaladas.
2. **Configuración del controlador de Chrome:** Utiliza ``webdriver_manager`` para gestionar la instalación del controlador de Chrome.
3. **Búsqueda en Amazon:**
 - Abre Amazon México.
 - Espera a que el cuadro de búsqueda esté disponible y realiza la búsqueda de "Discos duros SSD".
4. **Filtrado y ordenamiento:**
 - Filtra los resultados por la marca "ADATA".
 - Ordena los resultados por menor precio.
5. **Mantener el navegador abierto:** Utiliza ``input()`` para mantener el navegador abierto hasta que el usuario presione una tecla.
6. **Manejo de errores:** Si ocurre un error, se imprime en la consola.
7. **Cierre del navegador:** Al final, se cierra el navegador después de que el usuario presiona una tecla.

 El código al ser ejecutado marca el siguiente error "File "C:\Users\leone\PyCharm\Codigos\M2.2 Web scraping\.venv\Lib\site-packages\selenium\webdriver\common\driver_finder.py", line 56, in _binary_paths
browser = self._options.capabilities["browserName"]
^^
AttributeError: 'str' object has no attribute 'capabilities'" corrígelo



utiliza `webdriver.ChromeOptions()` para configurar el controlador de Chrome correctamente:

python

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

# Configuración del controlador de Chrome
options = webdriver.ChromeOptions()
driver = webdriver.Chrome(ChromeDriverManager().install(), options=options)

try:
    # Abre Amazon México
    driver.get("https://www.amazon.com.mx")

    # Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
    search_box.send_keys("Discos duros SSD")
    search_box.submit()

    # Espera a que aparezcan los resultados
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
    )

    # Filtra por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()

    # Espera a que se realice la selección de la marca
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
    )

    # Abre el menú de ordenamiento
    sort_dropdown = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
    )
    sort_dropdown.click()

    # Selecciona ordenar por menor precio
    sort_low_to_high = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "s-result-sort-select_1"))
    )
    sort_low_to_high.click()

    # Mantén el navegador abierto
    print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")

    # Espera a que el usuario presione una tecla para cerrar
    input("Presiona cualquier tecla para cerrar el navegador...")

except Exception as e:
    print(f"Ocurrió un error: {e}")

finally:
    # Cierra el navegador
    driver.quit()
```

Explicación:



- **Argumento `executable_path`:** Usé `executable_path` para pasar el ejecutable del controlador.
- **Manejo de errores:** Manejé cualquier excepción y proporcioné un mensaje en la consola.



Aquí tienes el código corregido sin el argumento `executable path`:

python

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

# Configuración del controlador de Chrome
options = webdriver.ChromeOptions()
driver = webdriver.Chrome(ChromeDriverManager().install(), options=options)

try:
    # Abre Amazon México
    driver.get("https://www.amazon.com.mx")

    # Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
    search_box.send_keys("Discos duros SSD")
    search_box.submit()

    # Espera a que aparezcan los resultados
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
    )

    # Filtra por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()

    # Espera a que se realice la selección de la marca
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
    )

    # Abre el menú de ordenamiento
    sort_dropdown = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
    )
    sort_dropdown.click()

    # Selecciona ordenar por menor precio
    sort_low_to_high = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "s-result-sort-select_1"))
    )
    sort_low_to_high.click()

    # Mantén el navegador abierto
    print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")

    # Espera a que el usuario presione una tecla para cerrar
    input("Presiona cualquier tecla para cerrar el navegador...")

except Exception as e:
    print(f"Ocurrió un error: {e}")

finally:
    # Cierra el navegador
    driver.quit()
```

Si el problema persiste, asegúrate de tener las versiones correctas de `selenium` y `webdriver_manager`. Aquí hay un script para verificar e instalar las versiones necesarias:

```
sh
```

```
pip install selenium==4.0.0
pip install webdriver-manager==3.4.2
```

Después de ejecutar esto, intenta ejecutar el código nuevamente. Esto debería resolver el problema y permitir que el script se ejecute correctamente, manteniendo el navegador abierto hasta que se presione una tecla.



sigue sin funcionar, creo que debes agregar "service" del webdriver



Memoria actualizada

Para resolver el problema, puedes usar la clase `Service` del módulo `selenium.webdriver.chrome.service` para configurar el controlador de Chrome. Aquí está el código actualizado con el uso de `Service`:

```
python
```

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

# Configuración del servicio de ChromeDriver
service = Service(ChromeDriverManager().install())

# Configuración del controlador de Chrome con opciones
options = webdriver.ChromeOptions()
driver = webdriver.Chrome(service=service, options=options)

try:
    # Abre Amazon México
    driver.get("https://www.amazon.com.mx")

    # Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
    search_box.send_keys("Discos duros SSD")
    search_box.submit()

    # Espera a que aparezcan los resultados
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
    )

    # Filtra por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()

    # Espera a que se realice la selección de la marca
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
    )
```

```

)

# Abre el menú de ordenamiento
sort_dropdown = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "a-autoid-0-announce")))
)
sort_dropdown.click()

# Selecciona ordenar por menor precio
sort_low_to_high = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "s-result-sort-select_1")))
)
sort_low_to_high.click()

# Mantén el navegador abierto
print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")

# Espera a que el usuario presione una tecla para cerrar
input("Presiona cualquier tecla para cerrar el navegador...")

except Exception as e:
    print(f"Ocurrió un error: {e}")

finally:
    # Cierra el navegador
    driver.quit()

```

Explicación de las correcciones:

1. **Uso de `Service`:** `Service` se usa para manejar el controlador de Chrome. Esto es una mejora que permite un manejo más limpio del controlador.
2. **Configuración del controlador:** `webdriver.Chrome()` se configura con `service=service` y `options=options`.

Con estos cambios, el código debería funcionar correctamente y mantener el navegador abierto hasta que el usuario presione una tecla. Si sigues teniendo problemas, asegúrate de que todas las dependencias estén actualizadas y que el entorno esté configurado correctamente.



YA FUNCIONA, ahora mejora el código para que no crea amazon que es un robot



Memoria actualizada

Para evitar que Amazon detecte y bloquee el script como un robot, puedes hacer algunos ajustes para simular el comportamiento humano de manera más realista. Aquí te muestro cómo mejorar el código para reducir la probabilidad de detección:

1. **Configurar el User-Agent:** Cambiar el User-Agent del navegador para simular un navegador real.
2. **Añadir pausas aleatorias:** Introducir pausas aleatorias para simular el comportamiento humano.
3. **Deshabilitar las extensiones del navegador:** Asegurarte de que el navegador no use extensiones que podrían ser detectadas.

Aquí está el código mejorado:

python

```

from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import random
import time

# Configuración del servicio de ChromeDriver
service = Service(ChromeDriverManager().install())

# Configuración del controlador de Chrome con opciones
options = webdriver.ChromeOptions()
options.add_argument("--disable-extensions") # Deshabilitar extensiones
options.add_argument("--disable-gpu") # Deshabilitar GPU
options.add_argument("--no-sandbox") # No usar sandbox
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36") # Cambiar el User-Agent

driver = webdriver.Chrome(service=service, options=options)

def human_delay():
    """Función para introducir una pausa aleatoria"""
    time.sleep(random.uniform(1.5, 3.5)) # Pausa entre 1.5 y 3.5 segundos

try:
    # Abre Amazon México
    driver.get("https://www.amazon.com.mx")
    human_delay()

    # Espera a que el cuadro de búsqueda esté presente y realiza la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
    search_box.send_keys("Discos duros SSD")
    human_delay() # Pausa antes de enviar la búsqueda
    search_box.submit()

    # Espera a que aparezcan los resultados
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
    )
    human_delay()

    # Filtra por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()
    human_delay()

    # Espera a que se realice la selección de la marca
    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[@aria-label='ADATA']"))
    )
    human_delay()

    # Abre el menú de ordenamiento
    sort_dropdown = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
    )
    sort_dropdown.click()
    human_delay()

    # Selecciona ordenar por menor precio
    sort_low_to_high = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "s-result-sort-select_1"))
    )
    sort_low_to_high.click()

```

```
human_delay()

# Mantén el navegador abierto
print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")

# Espera a que el usuario presione una tecla para cerrar
input("Presiona cualquier tecla para cerrar el navegador...")

except Exception as e:
    print(f"Ocurrió un error: {e}")

finally:
    # Cierra el navegador
    driver.quit()
```

Explicación:

1. **User-Agent:** Se ha configurado un User-Agent que simula un navegador real.
2. **Pausas aleatorias:** La función `human_delay()` introduce pausas aleatorias para hacer que el comportamiento sea menos predecible.
3. **Deshabilitar extensiones y GPU:** Las extensiones y la GPU se desactivan para evitar posibles señales de automatización.

Estos cambios ayudan a hacer que el comportamiento del navegador sea más parecido al de un usuario humano, reduciendo la posibilidad de que Amazon detecte la automatización.



El programa se ejecuta y despues de un tiempo manda el siguiente error "Ocurrió un error:
Message:
Stacktrace:

```
GetHandleVerifier [0x00438923+23283]
(No symbol) [0x003FE934]
(No symbol) [0x00330733]
(No symbol) [0x0037326F]
(No symbol) [0x003734AB]
(No symbol) [0x003AEE42]
(No symbol) [0x00394464]
(No symbol) [0x003ACB8D]
(No symbol) [0x003941B6]
(No symbol) [0x00368017]
(No symbol) [0x0036890D]
GetHandleVerifier [0x0052A5E3+1013683]
GetHandleVerifier [0x00533E3C+1052684]
GetHandleVerifier [0x0052D4A4+1025652]
GetHandleVerifier [0x0045EA2B+179195]
(No symbol) [0x00406833]
(No symbol) [0x00403198]
(No symbol) [0x00403337]
(No symbol) [0x003FB4BE]
BaseThreadInitThunk [0x77057BA9+25]
```

```
RtlInitializeExceptionChain [0x77BAC10B+107]
RtlClearBits [0x77BAC08F+191]"
```



El error que estás viendo es genérico y suele estar asociado con problemas en la comunicación entre Selenium y el navegador. Puede deberse a varias razones, como elementos no encontrados, tiempos de espera insuficientes, o problemas con la configuración del navegador o del controlador. Aquí tienes algunas estrategias para solucionar el problema:

1. Aumentar los Tiempos de Espera

Asegúrate de que los tiempos de espera sean suficientes para que los elementos se carguen correctamente. Puedes aumentar el tiempo de espera en `WebDriverWait``.

python

```
# Aumentar el tiempo de espera
search_box = WebDriverWait(driver, 20).until(
    EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
)
```

2. Captura de Pantalla en Caso de Error

Para obtener más detalles sobre el estado del navegador en el momento del error, puedes capturar una captura de pantalla.

python

```
finally:
    # Toma una captura de pantalla antes de cerrar el navegador
    driver.save_screenshot('screenshot.png')
    driver.quit()
```

3. Revisar los Selectores

Verifica que los selectores utilizados para localizar los elementos (como ``By.ID`` y ``By.XPATH``) sean correctos y que los elementos realmente existan en la página.

4. Uso de ``try-except`` en Bloques Específicos

En lugar de capturar todas las excepciones en un solo bloque ``except``, puedes envolver bloques específicos de código con ``try-except`` para identificar mejor dónde ocurre el problema.

python

```
try:
    # Código para realizar la búsqueda
    search_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
    )
```

```
search_box.send_keys("Discos duros SSD")
search_box.submit()
except Exception as e:
    print(f"Error durante la búsqueda: {e}")

try:
    # Código para filtrar por la marca ADATA
    filter_brand = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
    )
    filter_brand.click()
except Exception as e:
    print(f"Error durante el filtro de marca: {e}")

# Repetir para otras secciones del código
```

5. Actualizar Selenium y ChromeDriver

Asegúrate de tener las versiones más recientes de `selenium` y `chromedriver`. Puedes actualizar Selenium y el ChromeDriver con los siguientes comandos:

```
sh

pip install --upgrade selenium
pip install --upgrade webdriver-manager
```

6. Revisar Logs del Navegador

Puedes revisar los logs del navegador para obtener más información sobre los errores:

```
python

from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

capabilities = DesiredCapabilities.CHROME
capabilities['loggingPrefs'] = {'browser': 'ALL'}

driver = webdriver.Chrome(service=service, options=options,
desired_capabilities=capabilities)
```

Luego puedes acceder a los logs del navegador:

```
python

logs = driver.get_log('browser')
for log in logs:
    print(log)
```

Ejemplo de Código Mejorado:

Aquí tienes el código mejorado con algunos de estos ajustes:

python

```

from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import random
import time

# Configuración del servicio de ChromeDriver
service = Service(ChromeDriverManager().install())

# Configuración del controlador de Chrome con opciones
options = webdriver.ChromeOptions()
options.add_argument("--disable-extensions")
options.add_argument("--disable-gpu")
options.add_argument("--no-sandbox")
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36")

driver = webdriver.Chrome(service=service, options=options)

def human_delay():
    """Función para introducir una pausa aleatoria"""
    time.sleep(random.uniform(1.5, 3.5))

try:
    driver.get("https://www.amazon.com.mx")
    human_delay()

    # Búsqueda
    try:
        search_box = WebDriverWait(driver, 20).until(
            EC.presence_of_element_located((By.ID, "twotabsearchtextbox"))
        )
        search_box.send_keys("Discos duros SSD")
        human_delay()
        search_box.submit()
    except Exception as e:
        print(f"Error durante la búsqueda: {e}")

    # Filtrado por marca
    try:
        WebDriverWait(driver, 20).until(
            EC.presence_of_element_located((By.CSS_SELECTOR, "div.s-main-slot"))
        )
        filter_brand = WebDriverWait(driver, 20).until(
            EC.presence_of_element_located((By.XPATH, "//span[text()='ADATA']"))
        )
        filter_brand.click()
        human_delay()
    except Exception as e:
        print(f"Error durante el filtro de marca: {e}")

    # Ordenar por menor precio
    try:
        sort_dropdown = WebDriverWait(driver, 20).until(
            EC.presence_of_element_located((By.ID, "a-autoid-0-announce"))
        )
        sort_dropdown.click()
        human_delay()
        sort_low_to_high = WebDriverWait(driver, 20).until(
            EC.presence_of_element_located((By.ID, "s-result-sort-select_1"))
        )
        sort_low_to_high.click()
        human_delay()
    except Exception as e:
        print(f"Error durante el ordenamiento: {e}")

    print("Navegador abierto. La búsqueda se completó y los resultados están ordenados por menor precio.")

```



```
input("Presiona cualquier tecla para cerrar el navegador...")

finally:
    driver.save_screenshot('screenshot.png') # Captura de pantalla en caso de error
    driver.quit()
```

Prueba estos ajustes y verifica si solucionan el problema. Si el error persiste, proporciona más detalles sobre el contexto en el que ocurre para que podamos diagnosticarlo con más precisión.

ChatGPT puede cometer errores. Comprueba la información importante.