

## **Análisis de Librería Selenium en Python, para proceso de Web Scraping**

¿Qué es Selenium?

Es una "Suite de herramientas para automatización de navegadores", que también es asociada a testing o web scraping.

Selenium "No es una herramienta de testing o web scraping", se puede utilizar para esas tareas, pero su desempeño en las mismas puede no ser el óptimo.

La librería Selenium es una herramienta que nos permite utilizar a un navegador web simulando como si fuese un humano. Nos permite automatizar navegadores web.

Esta librería es muy usada cuando queremos obtener información de una página web que contiene JS y/o que requiera interacción humana, o en el caso, de simular un comportamiento humano.

Es la única librería que da visibilidad al JavaScript y, por lo tanto, podremos acceder a él y recoger la información.

Selenium necesita un navegador web instalado en el sistema para poder funcionar. Es compatible con navegadores como Firefox, Internet Explorer, Google Chrome, Opera, Safari. También es compatible con distintos lenguajes de programación como JAVA, C# Ruby, Python, PHP, JavaScript, PERL y Kotlin. Además de esto, también es necesario disponer un «webdriver» que permita manejar el navegador. Cada navegador tiene asociado un tipo de «driver». El «driver» es el manejador de las peticiones del usuario. Se trata del objeto fundamental en Selenium que nos permitirá interactuar con el navegador y los sitios web.

Cuando inicializamos el «driver» podemos asignar ciertas «capacidades» al navegador. Las podemos dividir en dos secciones: opciones y perfil.

- Una de las opciones más utilizadas es la capacidad de ocultar la ventana del navegador. Esto es útil cuando ya hemos probado que todo funciona y queremos automatizar la tarea.

- Es posible definir un perfil personalizado para usarlo en el navegador controlado por el «driver». Como ejemplo, podríamos querer desactivar javascript en el navegador (por defecto está activado).

Sin duda, es una librería muy potente. No obstante, es una librería que consume muchos recursos y un proyecto complejo, podría estar mal optimizado si no tienes un gran dominio de la librería.

Las características clave de Selenium son:

1. Puede trabajar fácilmente con Javascript (DOM)
2. Puede manejar fácilmente solicitudes AJAX y PJAX.

Una vez que hemos accedido a un sitio web, estamos en disposición de localizar elementos dentro del DOM. El objeto driver nos ofrece las siguientes funciones para ello:

Acceso	Función	Localizador
Clase	<code>find_elements_by_class_name()</code>	<code>By.CLASS_NAME</code>
Selector CSS	<code>find_elements_by_css_selector()</code>	<code>By.CSS_SELECTOR</code>
Atributo ID	<code>find_elements_by_id()</code>	<code>By.ID</code>
Texto en enlace	<code>find_elements_by_link_text()</code>	<code>By.LINK_TEXT</code>
Texto en enlace (parcial)	<code>find_elements_by_partial_link_text()</code>	<code>By.PARTIAL_LINK_TEXT</code>
Atributo NAME	<code>find_elements_by_name()</code>	<code>By.NAME</code>
Nombre de etiqueta	<code>find_elements_by_tag_name()</code>	<code>By.TAG_NAME</code>
XPath	<code>find_elements_by_xpath()</code>	<code>By.XPATH</code>

Todas estas funciones tienen su correspondiente versión para devolver un único elemento que cumpla con el filtro especificado. En caso de que hayan varios, sólo se devolverá el primero de ellos. El nombre de estas funciones sigue el patrón en singular:

`find_element_by_<accesor>()`

Si bien el acceso a la información de un sitio web puede ser un objetivo en sí mismo, para ello podríamos usar herramientas como Requests. Sin embargo, cuando entra en juego la interacción con los elementos del DOM, necesitamos

otro tipo de aproximaciones. Selenium nos permite hacer clic en el lugar deseado, enviar texto por teclado, borrar una caja de entrada o manejar elementos de selección, entre otros.

### **Selenium Tool Suite**

Selenium comprende un gran conjunto de herramientas o componentes.

**Selenio IDE:** Puede automatizar las pruebas de sus aplicaciones web mediante el uso de un componente de automatización de grabación y reproducción llamado Selenium IDE. Viene en forma de Firefox y Extensión del navegador Chrome. Las personas con poca y limitada experiencia en programación pueden utilizar esta herramienta, ya que funciona según los principios de grabación y reproducción.

**Selenio RC:** Selenium RC se utiliza para insertar código JavaScript en cualquier navegador para la automatización. También requiere un servidor adicional para ejecutar esos scripts de automatización. Pero tiene algunas limitaciones, como que es lento, no tiene un navegador autónomo como HtmlUnitDriver y es necesario iniciar el servidor antes de ejecutar cualquier script de prueba.

**Selenio WebDriver:** Selenium Webdriver es uno de los componentes más importantes de Selenium. Ofrece varios controladores para múltiples navegadores y admite diferentes lenguajes de programación. Además, está orientado a objetos y es compatible con casi todos los navegadores, como Firefox, Chrome, IE, Safari y más. La creación de secuencias de comandos se puede realizar mediante el uso de lenguajes populares, incluidos Python, Ruby, Java y más.

**Rejilla de Selenio:** La red de Selenium vuelve a ser una parte esencial de Selenium. Ayuda a distribuir la ejecución de pruebas automatizadas en paralelo entre diferentes máquinas remotas. El Grid consta de un concentrador y varios nodos. Estos nodos ejecutan las instancias de la suite Selenium en las que puede ejecutar casos de prueba. El concentrador central actúa como un servidor y los nodos están conectados a él para controlar toda la ejecución de la prueba.

## Características, pros y contras de Selenium WebDriver:

### Pros:

- Soporte para múltiples lenguajes.
- Fácil de instalar.
- Comunicación directa con el navegador (No requiere de un servidor intermedio).
- Interacción más realista (Como pensar que hace una persona y trasladarlo a Selenium).

### Contras:

- No soporta nuevos navegadores tan rápido o nuevas actualizaciones.
- No genera reportes o resultados de pruebas.
- Requiere de saber programar.

### Referencias:

- <https://geekflare.com/es/selenium-webdriver/>
- <https://github.com/jeffreyc/Curso-de-introduccion-a-Selenium-con-Python#Clase-2-Historia-de-Selenium>
- <https://eugdspace.eug.es/bitstream/handle/20.500.13002/913/TFG%20David%20Jimenez%20Lozano.pdf?sequence=1&isAllowed=y>
- <https://verneacademy.com/blog/articulos-data/web-scraping-paginas-dinamicas-selenium-python-y-beautifulsoup/>
- <https://aprendepython.es/pypi/scraping/selenium/>