BIG DATA

WEB SCRAPING

EDUARD LARA

1. INDICE

1. INTRODUCCION A WEB SCRAPING

- Actualmente la web tiene 1.5 millones de páginas las cuales en conjunto suman un total de 1.2 millones de terabytes de información.
- Es una cantidad ingente de datos, de diversa naturaleza desde lo más simple (ventas de productos) hasta lo mas complejo (chatGPT, Machine learning).
- Con tremenda cantidad de datos realizar la extracción manual es prácticamente imposible.
- **Web scraping es extraer datos de la web de una manera automática, utilizando programas realizados por nosotros mismos en el Python.**
- Podemos descargarnos todo lo que es visible a nuestros ojos cuando vemos una página web

1. VENTAJAS WEB SCRAPING

- Hay un pequeño número de páginas webs que nos proveen de mecanismos formales para descargar datos llamados APIs (por ejemplo Twitter, Foursquare, Google)
- **Web scraping no depende de APIs. Es independiente de si una plataforma provee o no de APIs para descargar su información.**
- **Web scraping permite descargar la información tal y como la vemos en nuestros navegadores**
- **Puede extraer datos de casi cualquier pagina web.**

1. DESVENTAJAs WEB SCRAPING

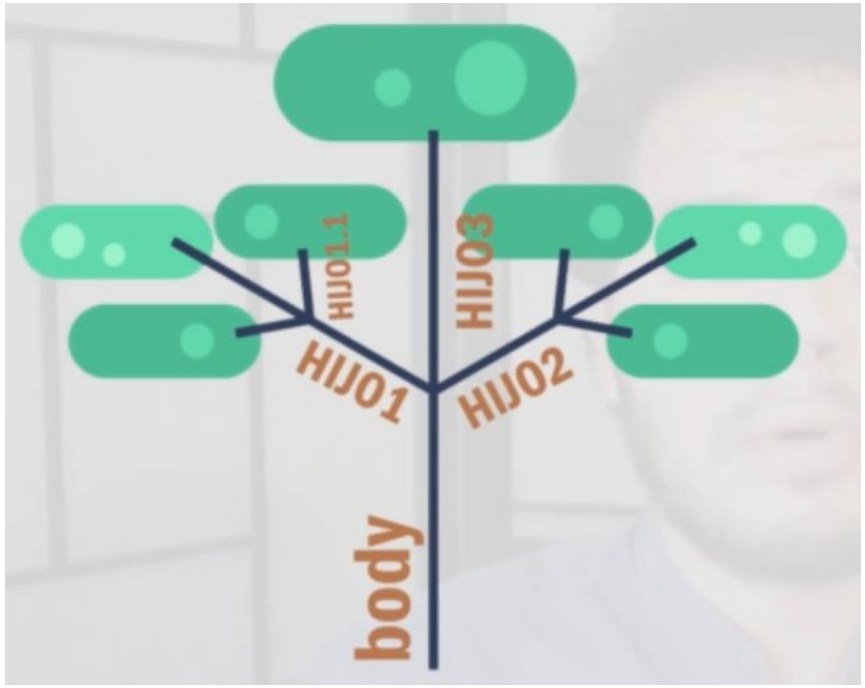
- Web scraping depende de la estructura visual de la misma.
- Si una página web se actualiza, debemos de actualizar nuestros programas para seguir extrayendo datos de manera correcta.
- Algunas paginas web pueden detectar nuestra actividad de extracción de datos y pueden prohibirnos el acceso (podemos caer en baneos temporales)

2. PAGINAS WEB: HTML

- El lenguaje de las paginas web es el HTML: Lenguaje de marcas utilizado para el desarrollo de paginas Internet
- Es importante tener **claro cómo se estructura un documento HTML para poder hacer web scraping.**
- **HTML es como si fuera un gran árbol** con muchas ramas donde cada rama tiene su propio nombre a los cuales vamos a llamar tags HTML o etiquetas HTML.
- La rama o tronco principal se llama Body. Este contiene varias ramas que pertenecen a el, las cuales llamaremos etiquetas hijos, los cuales a su vez tienen más hijos que están dentro de ellos

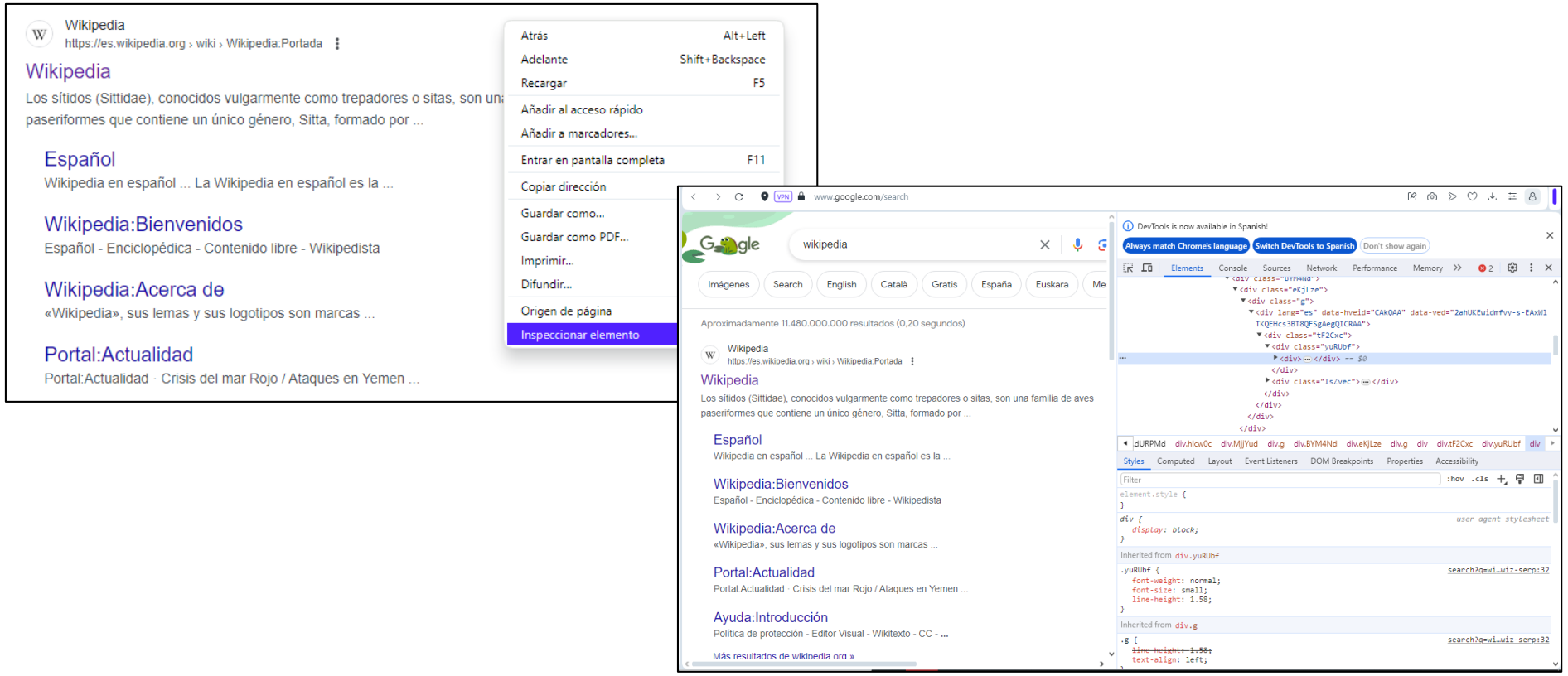
2. PAGINAS WEB: HTML

- En un documento HTML solo puede haber un Body, es el padre del resto de etiquetas.
- Sin embargo sus hijos y los hijos de sus hijos tienen nombres que se pueden repetir, no van a ser únicos



2. PAGINAS WEB: HTML

- Para ver el código HTML de una pagina solo debemos hacer click derecho y dar clic en Inspeccionar Elemento.
- Se abrirá una ventana al lado donde encontramos todos los elementos del HTML o DOM del HTML



3. ETIQUETAS HTML

Cada etiqueta o cada tag tiene una funcionalidad dentro del documento HTML.

- <p> → Contiene largos textos. Párrafos enteros
- <a> → Contiene links, URLS a otras páginas
- <button> → Contienen botones que embolsan acciones
- <form> → Contiene formularios
- <input> → Contiene cajitas de texto donde podemos escribir información. Usualmente es un hijo de form
- → Se utiliza como contenedor y en realidad puede contener cualquier cosa: texto, imágenes etc.
- → Para contener textos cortos de una línea

3. ETIQUETAS HTML

- <H1> <H2> <H3> <H4><H5> → contiene subtítulos de los títulos según va cambiando de importancia el título
- <img> → contiene imágenes
- → contiene listas de bullet points
- <table> → contiene tablas

Las etiquetas en si no son relevantes para web scraping. Lo que es relevante para web scraping es poder identificar la jerarquía de cada etiqueta es decir quién es hijo de quién

3. ETIQUETAS HTML

- En HTML cada etiqueta se abre y también se cierra:

    → Etiqueta de apertura y cierre.

- Lo que se encuentre dentro de las etiquetas de apertura y de cierre es el contenido o los hijos de esa etiqueta

```
<body>
  <div>
    <p>
      Hola Mundo
    </p>
  <div>
<body>
```

Body es el tronco principal, del cual sale una rama llamada div del cual a su vez sale otra rama llamada P la cual contiene una hoja con el mensaje "Hola mundo" que es el contenido visual que vemos

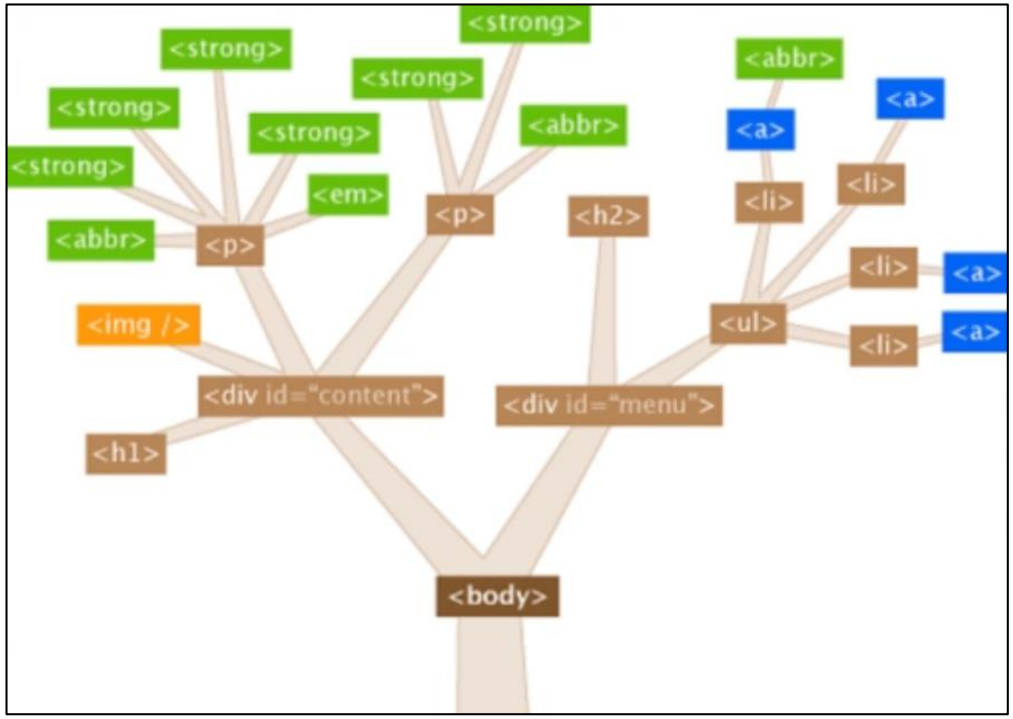
3. ETIQUETAS HTML

```
<body>
  <div>
    <p> Hola Mundo </p>
  </div>
  <span>
    Hola!
  </span>
</body>
```

Body tiene más de un hijo: y . Se dice que es una etiqueta hermano de la etiqueta <SPAN> porque están al mismo nivel

4. ATRIBUTOS ETIQUETAS HTML

- Los atributos de las etiquetas HTML son características adicionales a cada etiqueta, que nos permiten identificar o diferenciar las etiquetas que tiene el mismo nombre y se repiten a lo largo del árbol



4. ATRIBUTOS ETIQUETAS HTML

```
<div class="contenedor">
</div>
```

- Div tiene un atributo llamado Class que se compone de dos partes un nombre y el valor "contenedor"
- La etiqueta div ya la podemos identificar más fácilmente y diferenciarla del resto de etiquetas divs que existan a lo largo del árbol HTML.

4. ATRIBUTOS ETIQUETAS HTML

Los nombres de los atributos en realidad pueden ser cualquiera, sin embargo hay algunos estándares como:

- **Class** → Ayuda a darle estilo a las etiquetas.
- **Id** → Su valor tiene que ser único a lo largo de todo el HTML No pueden haber dos etiquetas que tengan el mismo valor del atributo Id
- **value** → Persiste lo que escribe un usuario dentro de una cajita de texto

5. TIPOS DE WEB SCRAPING

Nivel 1: Web scraping estático de una sola página. Cuando toda la información la tenemos en una sola página web y esta página no carga la información dinámicamente.

Vamos a utilizar las siguientes librerías

- **Requests** para realizar los requerimientos.
- **XML BeatifulSoup** para parsear los requerimientos HTML y extraer la información
- **Scrapy** que tiene ambas funcionalidades hacen los requerimientos y parsea los HTML respuestas para extraer la información.
- Descargaremos información de stackoverflow, Wikipedia, etc

5. TIPOS DE WEB SCRAPING

Nivel 2: Web scraping en estático de varias páginas del mismo dominio, llamado Scrolling Horizontal y Vertical.

Si buscamos un producto en ebay nos aparece la paginación de las opciones divididos entre varias páginas.

Scrolling horizontal → Cuando mi web scraping va recorriendo todas estas páginas donde se listan cada uno de los ítems o también llamado paginación.

Scrolling vertical → Es ir a las páginas del detalle de cada uno de los ítems de este listado.

- Scrapy. Sus librerías nos permite definir reglas para viajar a través de diversas páginas dentro de un mismo dominio
- Descargaremos datos de Airbnb, Trip Advisor.

5. TIPOS DE WEB SCRAPING

Nivel 3: Web scraping dinámico y consiste en automatizar las acciones de un navegador a través de programación (Python).

Controlaremos el navegador como si fuera un humano haciendo, click, scrolling, esperando por la carga de información etc. Y finalmente cuando se ha cargado la información que a esto se le llama carga dinámica realiza la extracción.

- Utilizaremos Selenium
- Descargaremos datos de Olx, Mercado Libre, Google Places y Twitter.

5. TIPOS DE WEB SCRAPING

Nivel 4: Web scraping de APIs

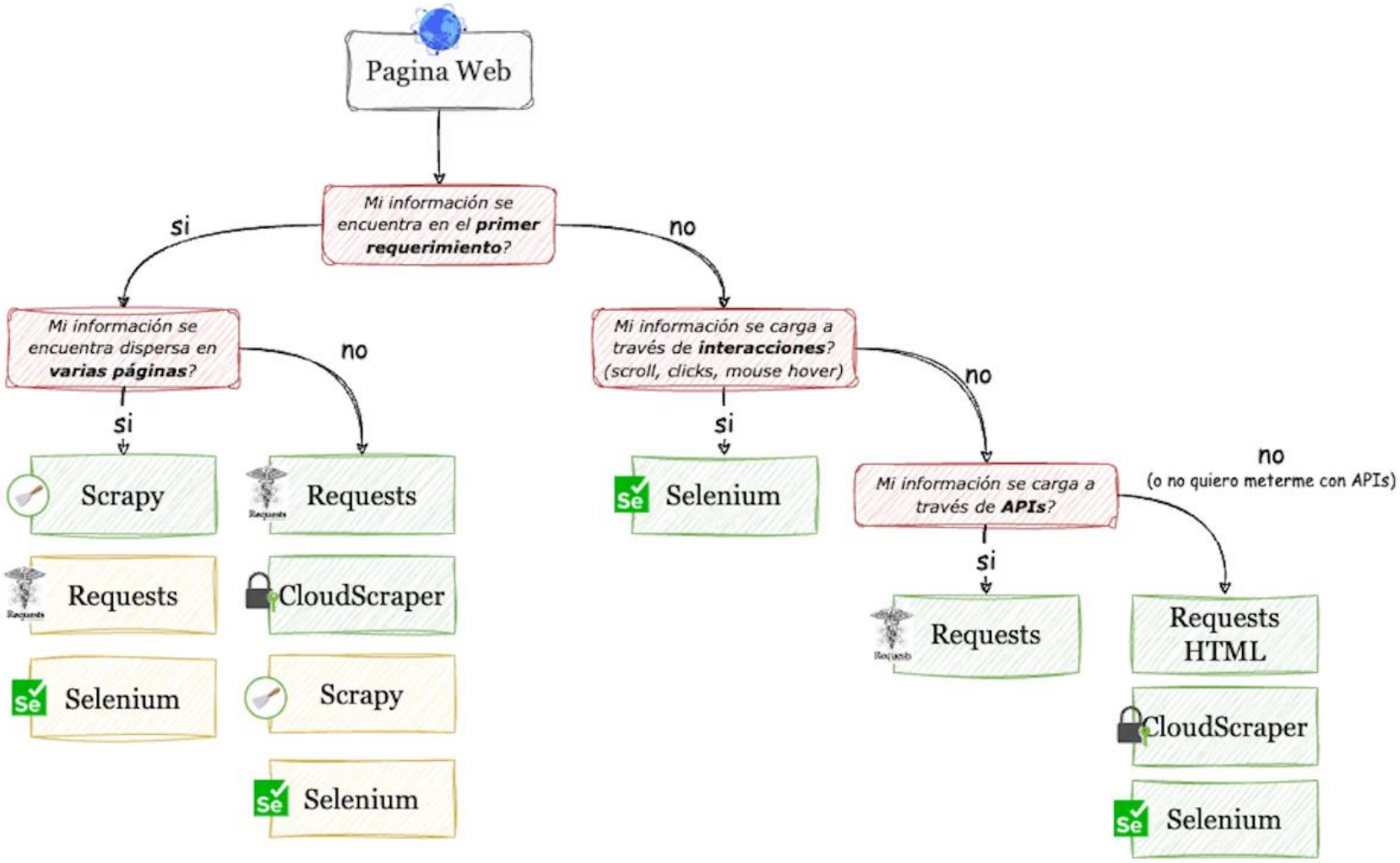
Nivel 5: A los tipos de web scraping existentes añadimos complejidad al vernos en la necesidad de realizar ciertas cosas como:

- Iniciar sesión, autenticarnos
- Llenar formularios
- Resolver Captchas
- Extraer información de iframes

6. HERRAMIENTAS WEB SCRAPING

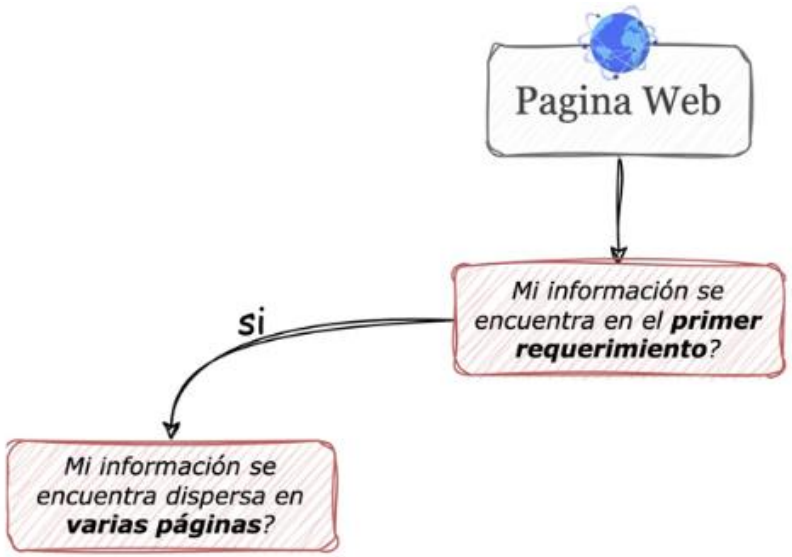
- Cada herramienta tiene su caso de uso.
- Hay mucha variedad de páginas web y cada herramienta es más eficiente que otras
- No existe una herramienta que sirva para todas las páginas webs,
- En ciertos casos sólo podremos utilizar una de ellas.
- Vamos a ver una Guia para reconocer que herramienta utilizar según la página web que tengamos enfrente.

6. HERRAMIENTAS WEB SCRAPING



6. HERRAMIENTAS WEB SCRAPING

- Cuando nos encontramos con una página web, lo primero que nos preguntamos es si es que toda la información que quiero extraer me llega con el primer requerimiento que hace la página web al servidor.
- De ser así, ahora tengo que preguntarme si mi información se encuentra en una sola página o está dispersa en varias páginas.



6. HERRAMIENTAS WEB SCRAPING

- Si datos dispersos en varias páginas → se recomienda utilizar Scrapy (permite viajar través de varias páginas)
- Se puede utilizar Requests y Selenium, pero no es lo más recomendable, es menos eficiente y más complicado implementar en código.

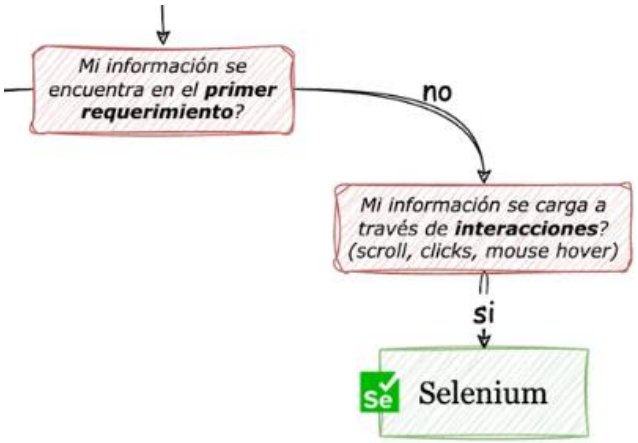
6. HERRAMIENTAS WEB SCRAPING

- Si los datos se encuentran solamente en una URL, lo recomendable es utilizar Requests o CloudScraper.

- CloudScraper es una herramienta que sirve para hacer requerimientos a páginas que tienen fuertes mecanismos de detección de bots.
- También podremos utilizar Scrapy o Selenium, pero no será lo más eficiente.

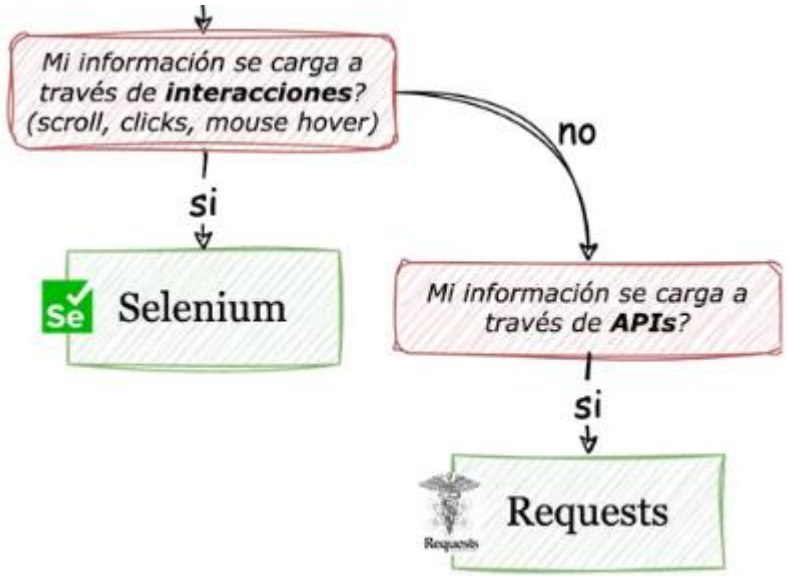
6. HERRAMIENTAS WEB SCRAPING

- Volviendo a la primera pregunta, si la respuesta es que mi información no llega con el primer requerimiento, entonces tendré que preguntarme si mi información se carga con alguna interacción humana en la página.
- Si la información se carga al hacer scrolling o al pasar el mouse por encima del elemento de la página, o por ejemplo, al dar click en algún elemento de la página.
- Si la respuesta es sí, Selenium es la única opción.



6. HERRAMIENTAS WEB SCRAPING

- Si la respuesta es no, tendré que preguntarme si mi información se carga a través de APIs, en cuyo caso request será la mejor alternativa.



6. HERRAMIENTAS WEB SCRAPING

- Si la información no se carga a través de APIs o en su defecto, no queremos meter con la complejidad que implica analizar una API, podemos utilizar Requests HTML, CloudScrapper o Selenium, siendo las tres igual de eficientes en este caso.