Modulo 1: Web Scraping

Relator: Felipe Mesa Abraham

Agenda

- Sesión 1:
  - Introducción a Web Scraping
  - Repaso de HTML
  - Actividad práctica con Request
  - Actividad práctica con Beautiful Soup

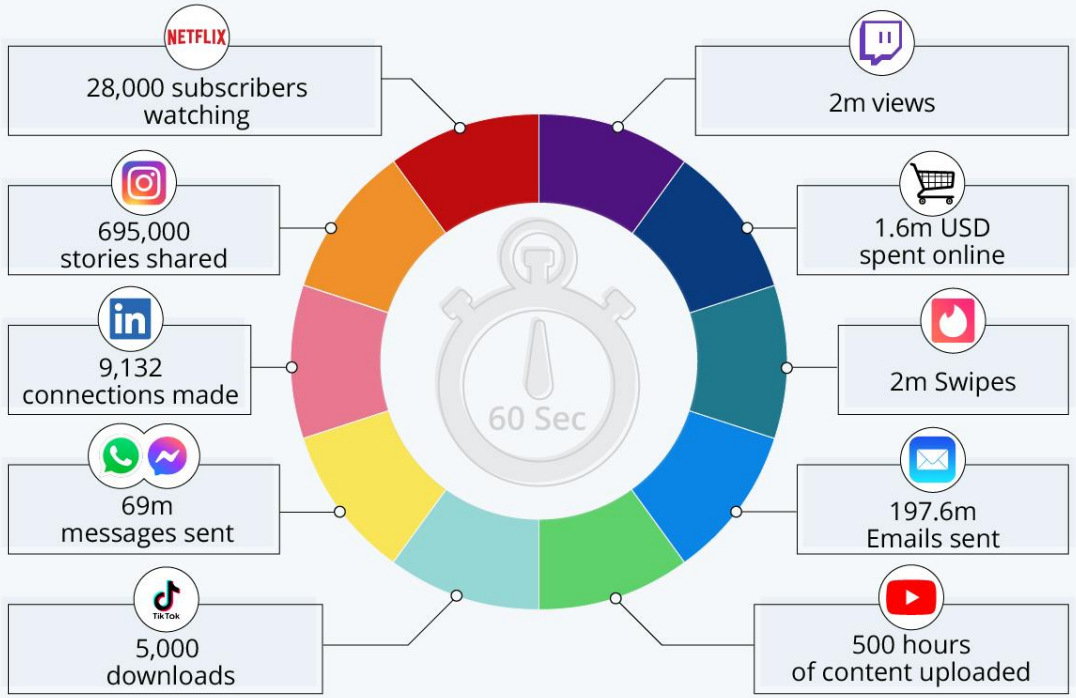
- Sesión 2:
  - Sintaxis de localizadores
  - Actividad práctica con Selenium
  - Actividad práctica con Scrapy

¿Que es Web Scraping?

- Es una técnica para extraer datos desde la web de manera eficiente.
- Consiste en analizar el código fuente de una página web para extraer información relevante.
- Permite extraer grandes volúmenes de datos para luego organizarlos y estructurarlos.

A Minute on the Internet in 2021

Estimated amount of data created on the internet in one minute



Source: Lori Lewis via AllAccess

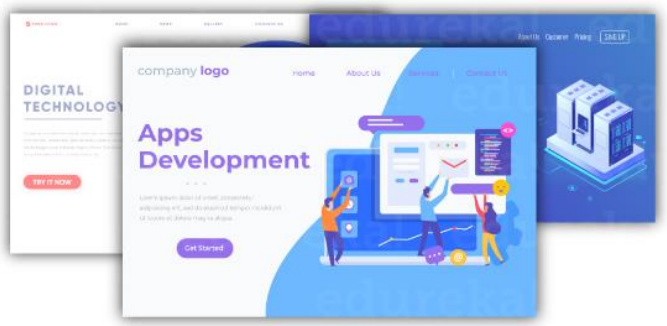
statista

¿Para que hacer Web Scraping?

- Monitoreo de precios
- Comparación de productos
- Estudios de mercado
- Análisis de sentimientos
- Análisis periodísticos
- Mucho mas!

Algunos Ejemplos Cotidianos

Proceso

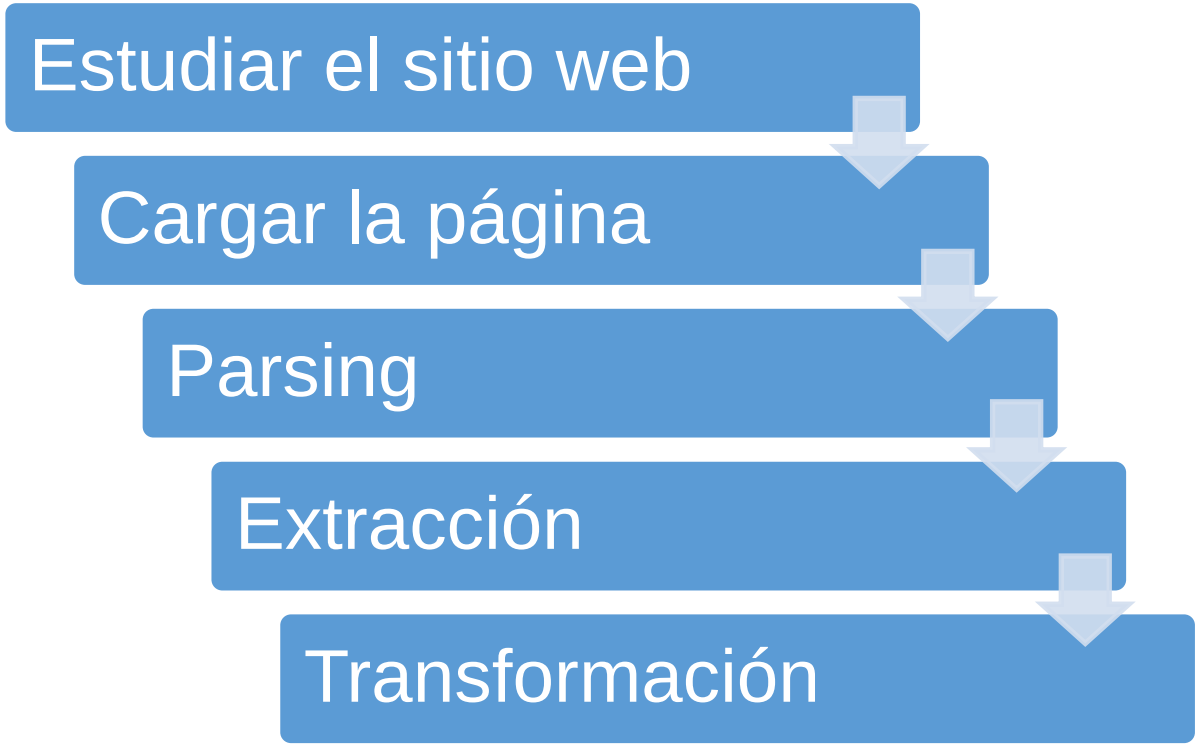


Webpages

Web Scraping

Structured Data

Proceso


Estudiar el sitio web
Cargar la página
Parsing
Extracción
Transformación

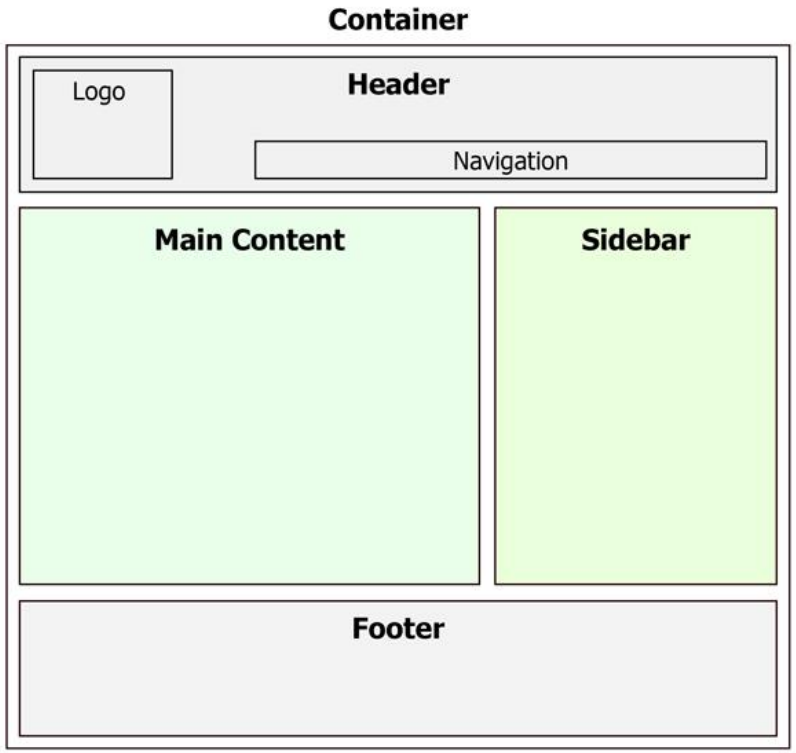
¿Como se estructura un sitio web?

- La mayoría de los sitios web están creados usando HTML, CSS y Javascript.
- No es necesario saber programar en estos lenguajes, pero si entender lo básico.

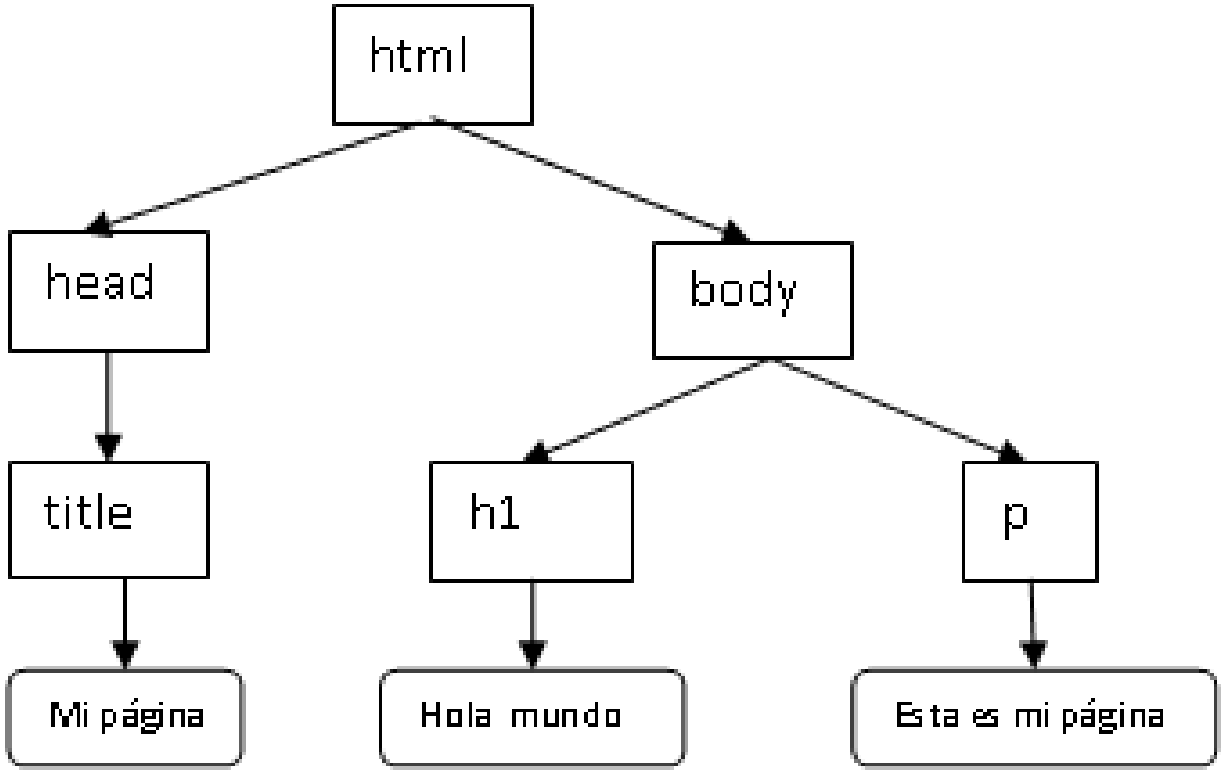
Estructura Estética Interactividad

¿Como se estructura un sitio web?

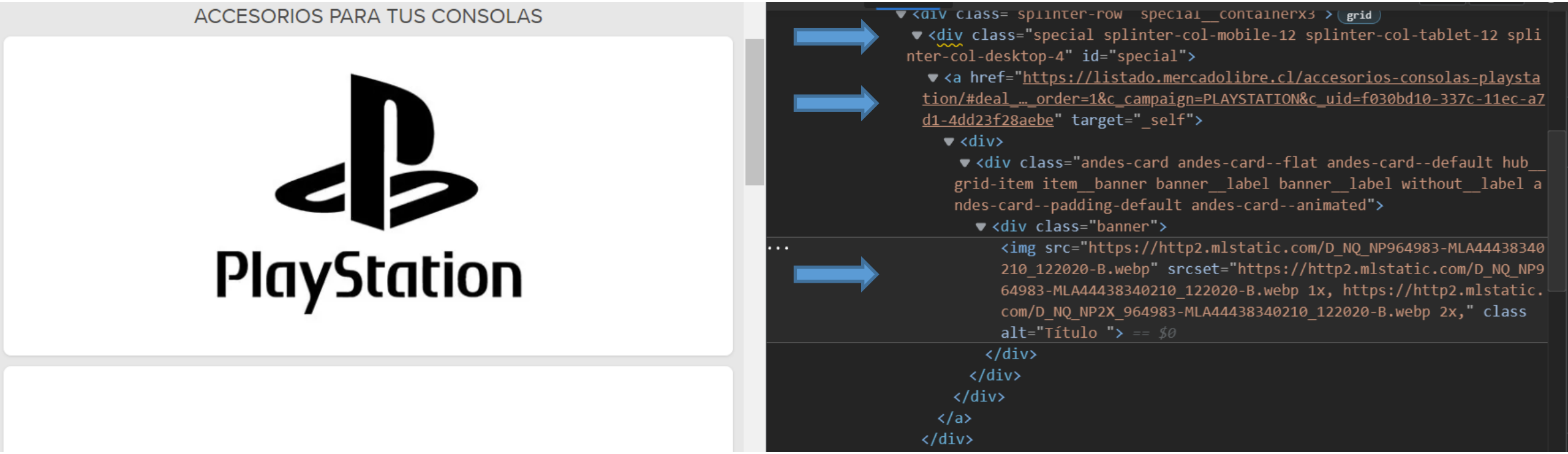
- Muchas “cajas” dentro de otras “cajas”.
- Todas estas “cajas” tienen una clase que las distingue y un estilo asignado.
- Estas “cajas” pueden ser interactivas.



¿Como se estructura un sitio web?



Ejemplo


ACCESORIOS PARA TUS CONSOLAS
PlayStation
grid

<img src="https://http2.mlstatic.com/D_NQ_NP964983-MLA44438340210_122020-B.webp" srcset="https://http2.mlstatic.com/D_NQ_NP964983-MLA44438340210_122020-B.webp 1x, https://http2.mlstatic.com/D_NQ_NP2X_964983-MLA44438340210_122020-B.webp 2x," class
alt="Título "> == $0

Breve repaso de HTML

- Etiquetas:
  - head: La parte superior del documento HTML.
  - body: Indica la parte del cuerpo del contenido de un documento HTML.
  - div: Un elemento que es usado mayoritariamente para agrupar otros elementos.
  - p: Etiqueta que nos sirve para agrupar texto dentro de un párrafo.
  - a: Es una etiqueta que nos ayuda a poder crear un enlace a una página web.
  - td: Etiqueta que permite crear tablas en una pagina web.

- Atributos:
  - id: Identificador único de un elemento.
  - class: Establece la clase de estilos CSS que se aplica que se aplican a un elemento.
  - href: Atributo que permite adjuntar enlaces en algún elemento de la página.

*https://www.w3schools.com/TAGS/default.ASP

Web Scraping – Paquetes en Python

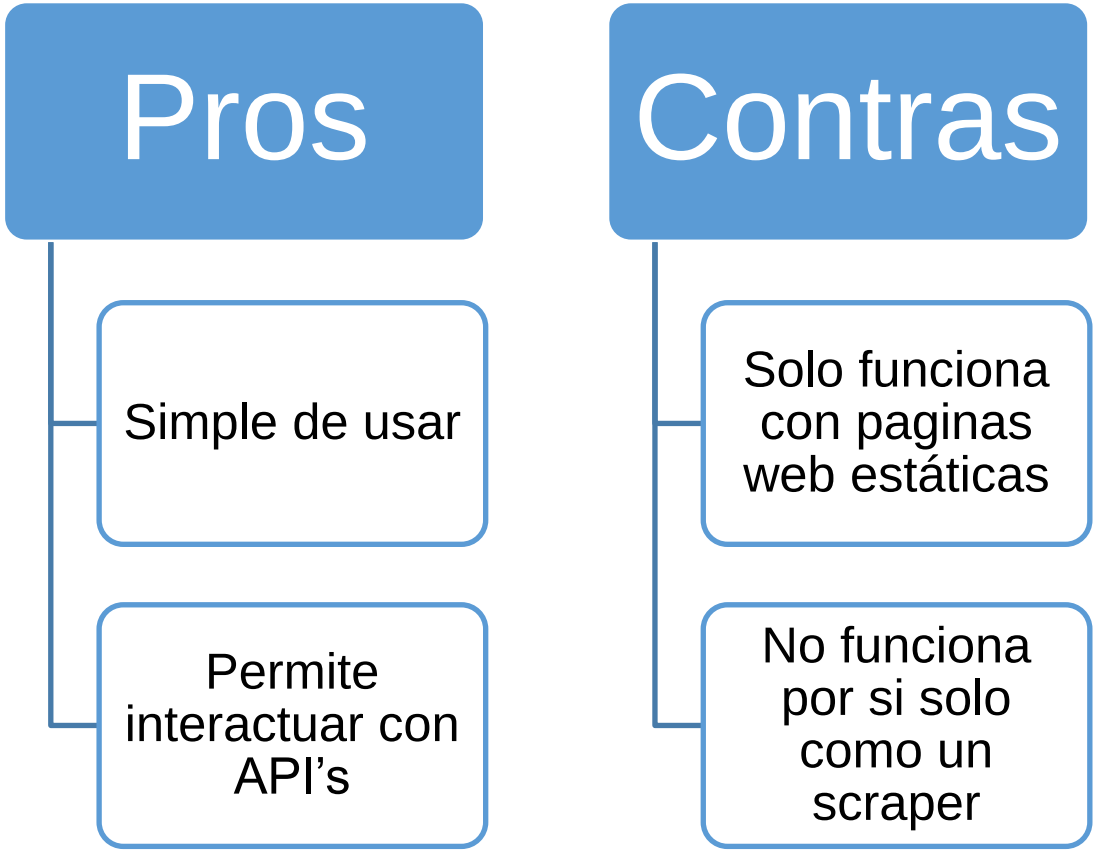
Requests

- Es una librería que permite hacer solicitudes HTTP de manera muy simple.
- Como respuesta, entrega el código fuente de la página solicitada.
- Permite también interactuar con API’s y extraer información de ellas.

- Notebook: [Web Scraping Requests.ipynb - Colaboratory (google.com)]

* Documentación: https://docs.python-requests.org/en/latest/

Requests

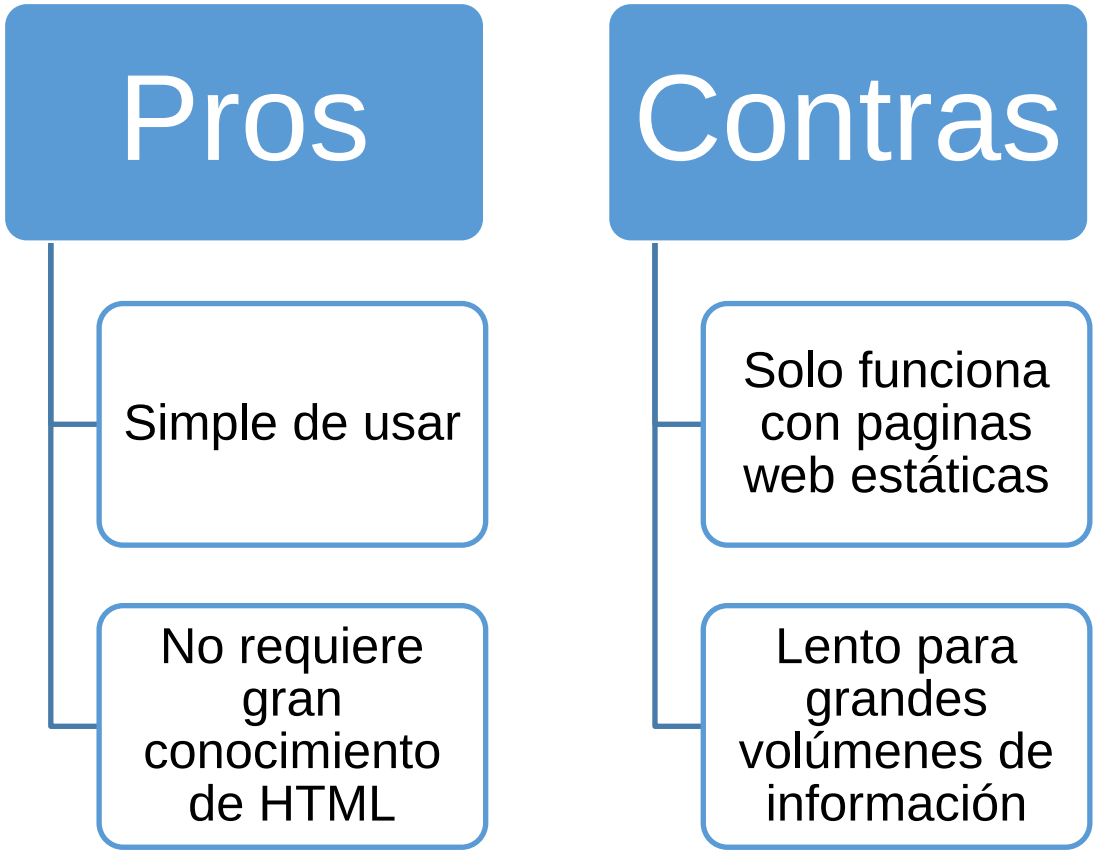

Pros
Simple de usar
Permite interactuar con API’s
Contras
Solo funciona con paginas web estáticas
No funciona por si solo como un scraper

Request y Beautiful Soup

- Mientras que Requests hace la solicitud HTTP, Beautiful Soup se encarga de extraer la información relevante.
- Beautiful Soup “parsea” el código fuente y extrae la información buscada.
- Notebook: Web Scraping Beautiful Soup.ipynb - Colaboratory (google.com)

*Documentación: https://beautiful-soup-4.readthedocs.io/en/latest/

Requests y Beautiful Soup

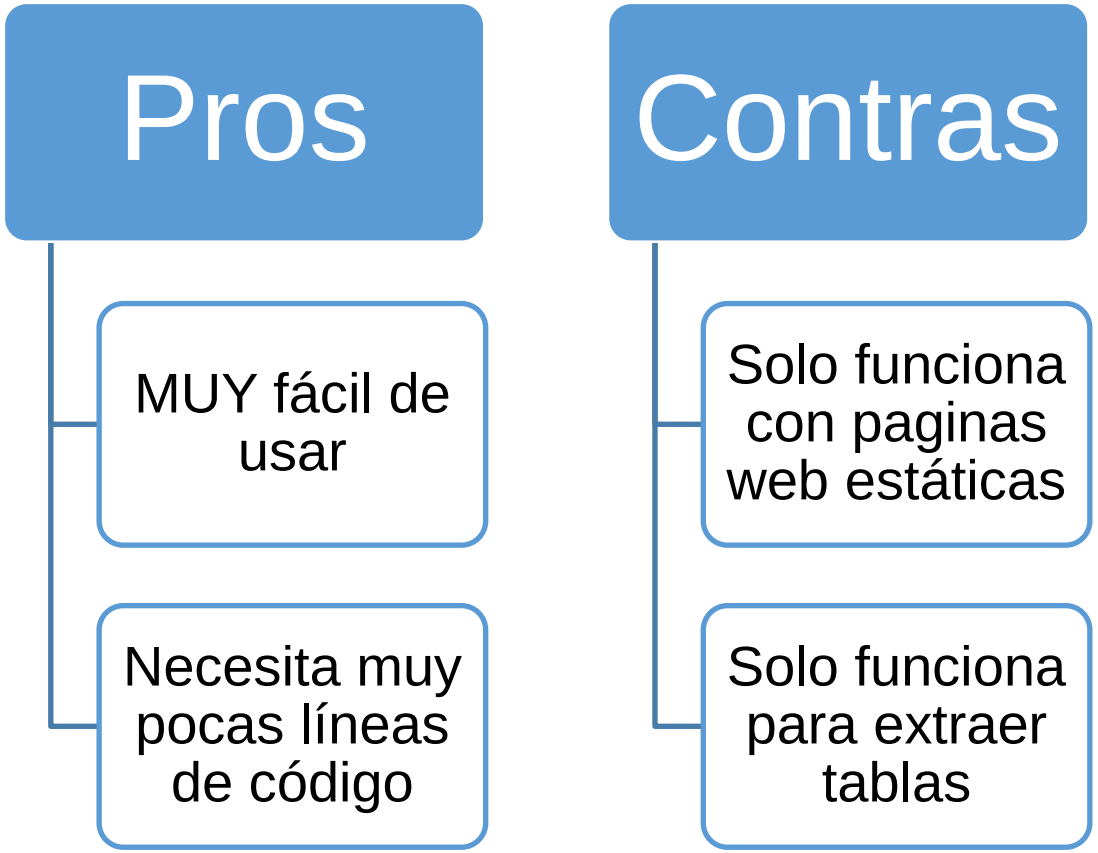


Pandas

- En casos muy particulares, Pandas permite extraer información de un pagina, con muy pocas líneas de código.
- Se utiliza la función read_html() para extraer las tablas que pueda tener una página.
- Notebook: Web Scraping Pandas.ipynb - Colaboratory (google.com)

*Documentación: https://pandas.pydata.org/docs/reference/api/pandas.read_html.html

Pandas


Pros
MUY fácil de usar
Necesita muy pocas líneas de código
Contras
Solo funciona con paginas web estáticas
Solo funciona para extraer tablas

Sintaxis de localizadores (locator syntax)

- Los localizadores son instrucciones para determinar la ubicación de elementos en una página web.
- Se basan en la estructura HTML de la página.

- Existen muchos tipos de localizadores:
  - Xpath
  - CSS Selector
  - id
  - class name
  - Etc.

Sintaxis de localizadores (locator syntax)

Localizador	Descripción
class name	Localiza elementos que contengan el nombre de la clase a buscar (no se permiten nombres de clase compuestos)
id	Localiza elementos cuyo atributo de ID coincide con el valor de búsqueda.
link text	Localiza elementos tipo <a> cuyo texto visible coincide con la búsqueda.
tag name	Localiza elementos cuyo nombre de etiqueta o tag coincide con la búsqueda.
css selector	Localiza elementos que coinciden con un CSS Selector.
xpath	Localiza elementos que coincidan con un xpath

*Documentación: https://www.selenium.dev/documentation/webdriver/locating_elements/

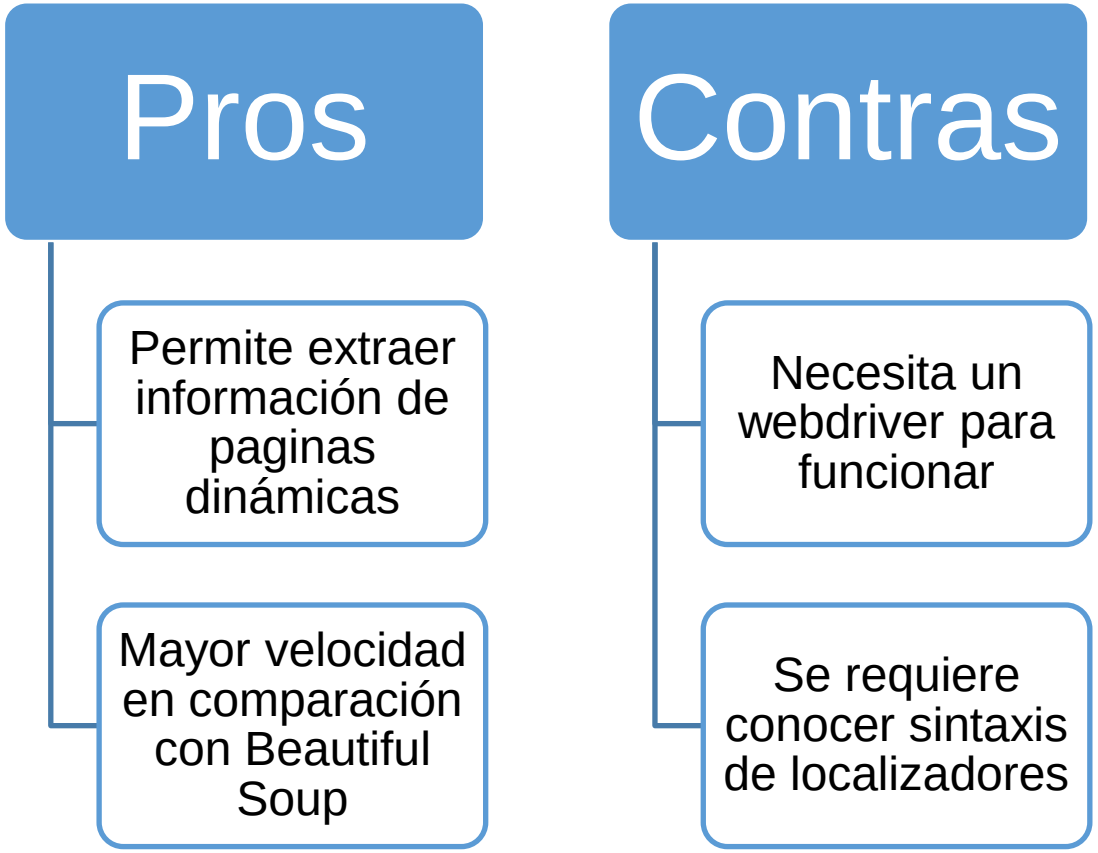
Selenium

- Permite realizar Web Scraping por medio de un driver.
- El driver simula el comportamiento de un usuario real usando un navegador.
- Se basa en el uso de localizadores para extraer información.

- Notebook: Web Scraping Selenium.ipynb - Colaboratory (google.com)

*Documentación: https://www.selenium.dev/documentation/

Selenium


Pros
Permite extraer información de paginas dinámicas
Mayor velocidad en comparación con Beautiful Soup
Contras
Necesita un webdriver para funcionar
Se requiere conocer sintaxis de localizadores

Xpath y CSS Selector

- Son un tipo especial de localizadores.
- Usan la estructura HTML para ubicar elementos.
- Consideran los nodos padres y nodos hijos.
- Estos localizadores son necesarios para extraer información usando Scrapy.

Xpath y CSS Selector

- Ejemplo: Seleccionar todos los <p> contenidos en los con la clase “clase1”
  - XPath: //div[@class=“clase1”]/p
  - CSS Selector: div.clase1 > p

- Ejemplo 2 seleccionar todos los títulos de los productos en la página: PlayStation | MercadoLibre.cl
  - Xpath: //h2[@class=“ui-search-item__title”]
  - CSS Selector: h2.ui-search-item__title

*Documentacion:
-CSS Selectors Reference (w3schools.com)
-XPath Operators (w3schools.com)

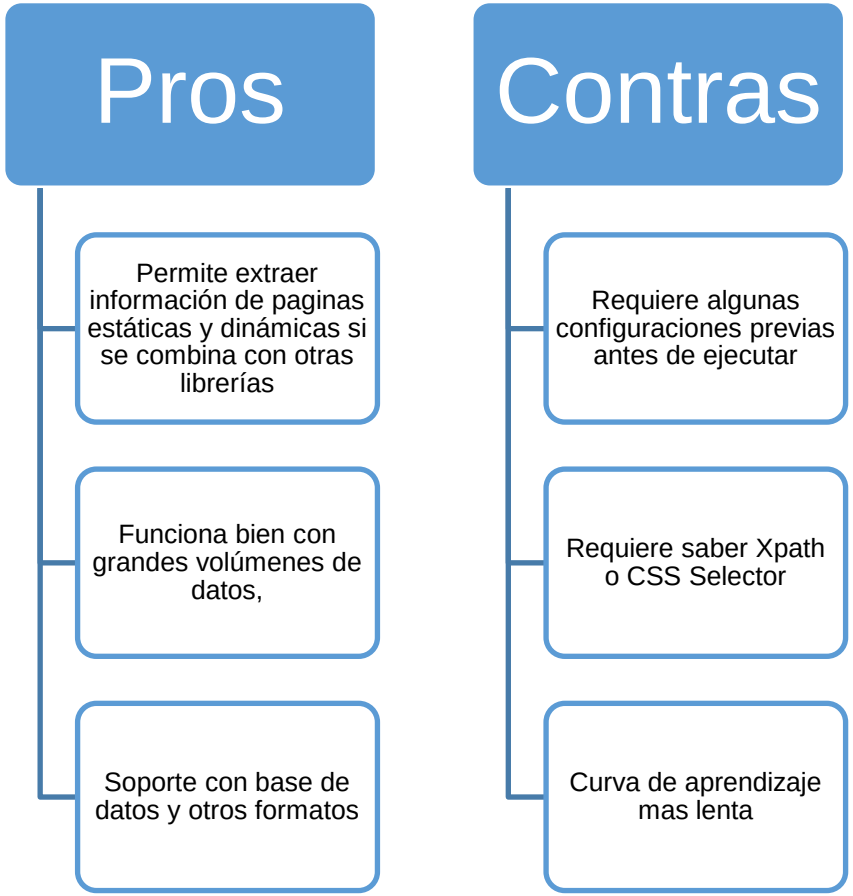
*Selector Gadget:
- SelectorGadget - Chrome Web Store (google.com)

Scrapy

- A diferencia de los otros paquetes que hemos revisado, Scrapy funciona como un framework.
- Permite extraer grandes volúmenes de datos haciendo muchas solicitudes.
- Mayor rapidez que otros paquetes.

*Documentación: https://docs.scrapy.org/en/latest/

Scrapy


Pros
Permite extraer información de paginas estáticas y dinámicas si se combina con otras librerías
Funciona bien con grandes volúmenes de datos,
Soporte con base de datos y otros formatos
Contras
Requiere algunas configuraciones previas antes de ejecutar
Requiere saber Xpath o CSS Selector
Curva de aprendizaje mas lenta

Buenas Prácticas

- Siempre estudiar bien la página web que se va a hacer scraping.
- Probar un localizador usando el navegador antes de incluirlo en el scraper.
- En caso que el scraper no funcione, probar con distintos localizadores, por ejemplo, un Xpath puede funcionar mejor que un CSS Selector.
- Usar “time sleep” en distintas partes del proceso puede ser conveniente.
- En caso de trabajar de manera local, crear un entorno virtual para evitar problemas de dependencias.

Lecturas de profundización

Serverless Architecture for a Web Scraping Solution | AWS Architecture Blog (amazon.com)

How to deploy python selenium script on Heroku – YouTube

Web Scraping Cloud Hosting & Data Extraction - Zyte

¡Gracias!