Laboratorio Nacional de Políticas Públicas

CIDE CENTRO DE INVESTIGACIÓN Y DOCENCIA ECONÓMICAS A.C.

Web Scraping

(Introducción)

Visualización y puesta en página web

Septiembre, 2020

M.C. JORGE JUVENAL CAMPOS FERREIRA.
Investigador Asociado.
Laboratorio Nacional de Políticas Públicas
CIDE

Hoja de Ruta.

1. Revisión de conceptos básicos.

2. Revisión de conceptos legales.

3. Librería y funciones de R.

4. Ejemplo práctico.

Conceptos Básicos

Concepto: Web Scraping

"Proceso de ***extracción de datos almacenados en la web.*** *Su objetivo es el de recopilar información almacenada en un servidor web. Podemos "escrapear" artículos web, e-commerce, obtener precios, reseñas, etc."*
(Platzi).

Concepto: Web Scraping

"***La práctica de recolectar datos de manera automatizada a través de internet***, *sin recurrir a la interacción con una API o al trabajo de un humano recolectando información*".

(Ryan Mitchell, Web Scraping with Python).

Concepto: Web Scraping

Activity where a party **uses automated software** to go and **crawl the internet and copy data and other content** so that can **compile it together** and make its **own product offering.**
(Evan Brown).

Discusión: ¿Qué tan Legal es el WS?

Si tu respuesta es “Si”, replantéate lo que piensas hacer

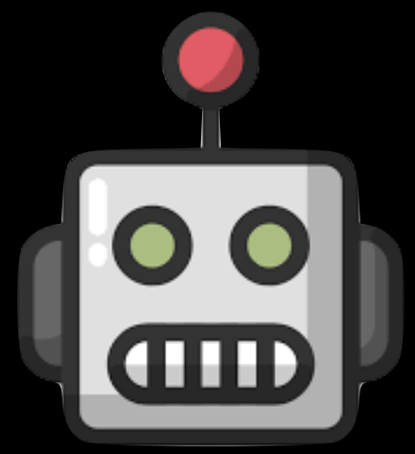
1. ¿Estoy violando alguna reglamentación local?
2. ¿Estoy violando los **Términos y Condiciones** del sitio?
3. ¿Estoy accediendo a lugares **no autorizados,** o a lugares donde se necesita hacer **login**?
4. ¿Es legal el uso que le voy a dar a los datos o **genera algún perjuicio de algún modo**?
5. ¿Estoy violando alguna ley de **derechos de autor**?
6. ¿Estoy **accediendo a datos personales**, o a datos que pudieran violar el **secreto comercial**?

Concepto: Robots.txt

El **robots.txt** es un archivo que define buenas prácticas a la hora de hacer scraping. Nos dice a qué sitios no quiere la página que accedamos.

Es recomendable seguir estas indicaciones, ya que no hacerlo puede ser **no-ético** y puede tener, en algunos casos**, consecuencias legales.**

https://platzi.com/robots.txt
https://www.linkedin.com/robots.txt
https://www.facebook.com/robots.txt
https://www.imdb.com/robots.txt
https://twitter.com/robots.txt



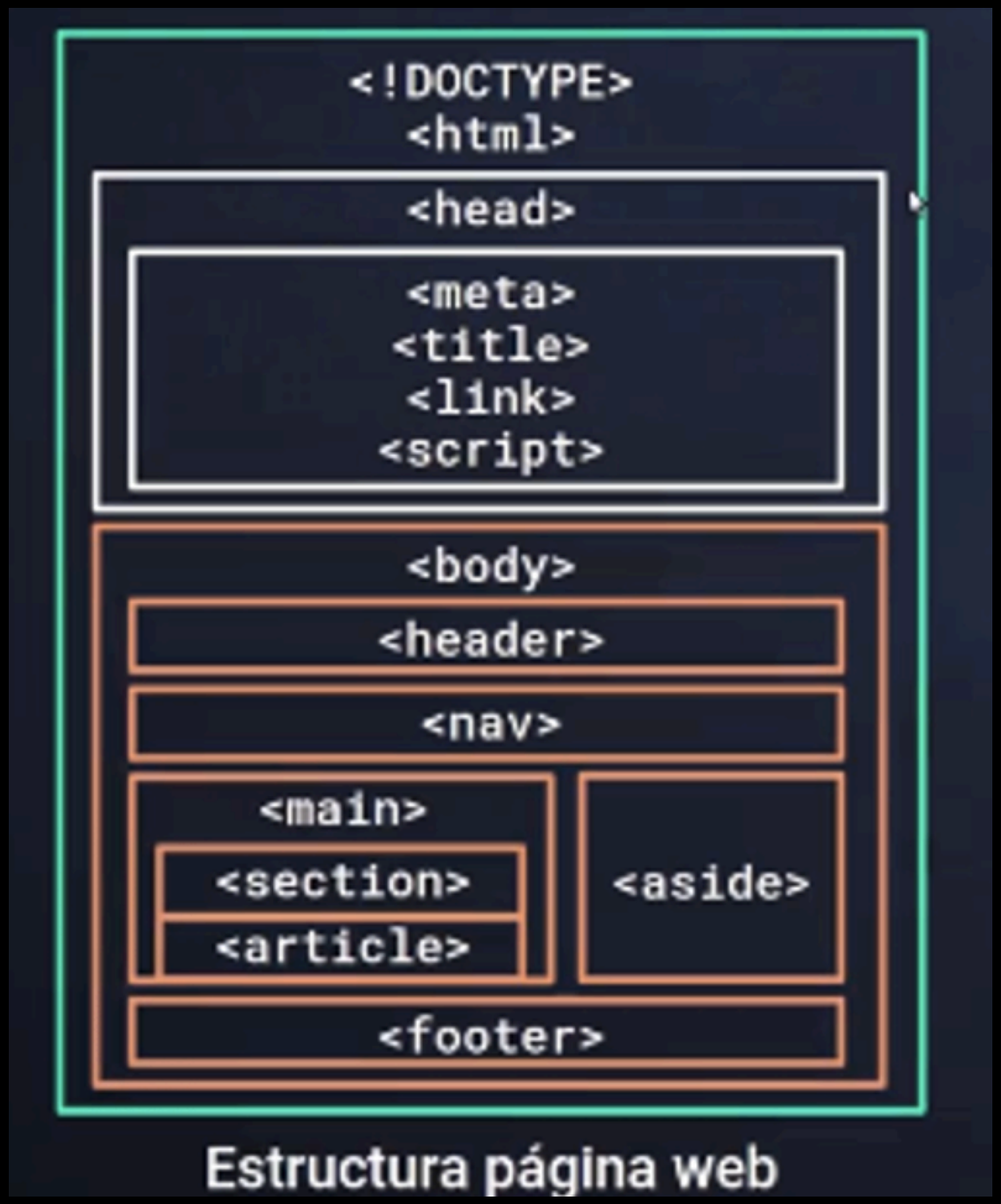
Páginas Web

Árbol

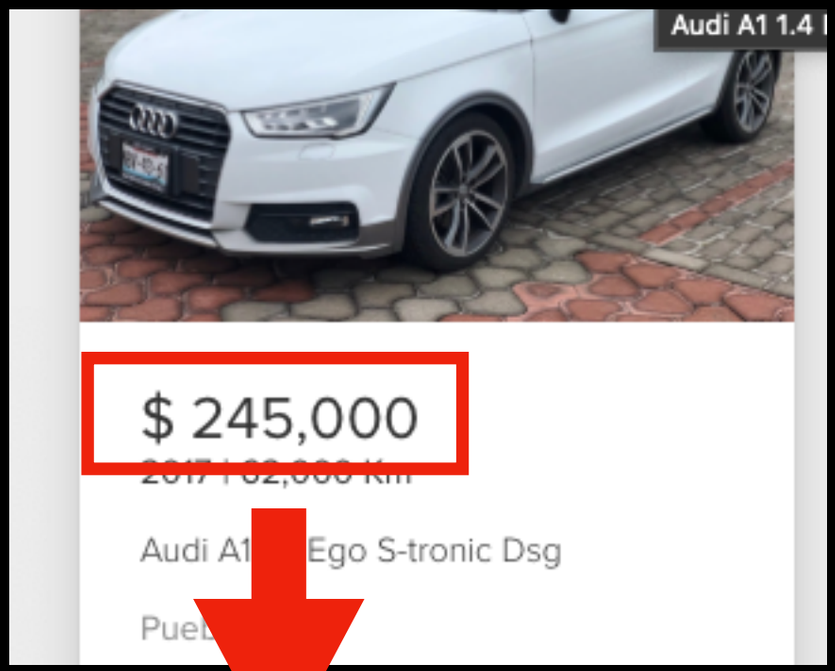
Inspector

```
<!DOCTYPE html>
▼ <html lang="es-mx">
  ▶ <head>…</head>
  ▼ <body>
    ▼ <div class="wrapper">
      ▼ <header class="header">
        ▼ <nav class="nav">
          ▼ <a href="/" class="nav-logo">
              <img src="/images/logo.png" width="50" height="50" alt=
              "Logo">
            </a>
          ▼ <ul class="nav-links">
            ▼ <li>
                <a href="/sobre_mi/">Sobre mí</a>
              </li>
            ▶ <li>…</li>
            ▶ <li>…</li>
            ▶ <li>…</li>
            </ul>
          </nav>
        </header>
      ▼ <section class="hero">
        ▼ <div class="hero-inner">
            <h1>Juve Campos</h1>
            <h2>Blog personal 😎 👨‍💻 </h2>
          </div>
        </section>
      ▼ <main class="content" role="main">
        ▼ <div class="archive">
            <h2 class="archive-title">2020</h2>
          ▼ <article class="archive-item">
              <a href="/2020/07/26/cortando-islas/" class="archive-item-
              link">Cortando Islas</a>
              <span class="archive-item-date"> 2020-07-26 </span>
            </article>
          ▼ <article class="archive-item">
              <a href="/2020/03/03/ejemplos-de-lo-que-se-puede-hacer-en-
              shiny/" class="archive-item-link">Ejemplos de lo que se
              puede hacer en shiny</a>
              <span class="archive-item-date"> 2020-03-03 </span>
            </article>
          ▶ <article class="archive-item">…</article> = $0
          ▶ <article class="archive-item">…</article>
          ▶ <article class="archive-item">…</article>
          ▶ <article class="archive-item">…</article>
          </div>
        </main>
      ▼ <footer class="footer">
        ▶ <ul class="footer-links">…</ul>
        </footer>
      </div>
    </body>
</html>
```

Páginas Web



Estructura página web



Funciones de R

Funciones de R

La librería rvest **nos ayuda a hacer web-scraping.**

Esta librería está **diseñada para trabajar con el tidyverse (%>%)** .

Trata de hacer **fácil las tareas más comunes de web-scraping**, inspirada en la librería beautifulsoup de Python.

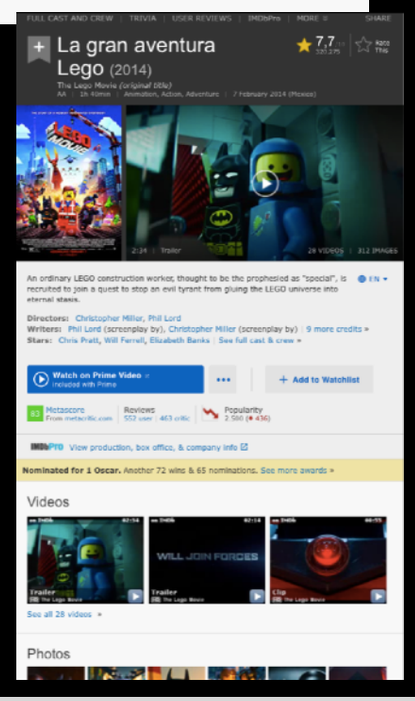
rvest::read_html(url)

Esta función nos permite descargar a nuestra sesión de R el código HTML de la página a la que le queremos sacar información.

```
library(rvest)

lego_movie <- read_html("http://www.imdb.com/title/tt1490017/")
```

Argumento: **url, cadena de texto.**
La dirección web de la página a la cual le queremos extraer información.



rvest::html_node()

Esta función extrae un nodo (rama) del código de la página web que tenemos almacenada en un objeto generado por *rvest::read_html()*

```
rating <- lego_movie %>%
  html_nodes("strong span")
```

**Atrapa los nodos del tipo "strong span" dentro del código html.*

El resultado de esta función aún no es legible. Hay que transformarlo (como lo que nos salía de las APIs).

rvest::html_text()

Esta función extrae el contenido que se encuentra dentro de un nodo (rama), resultante de la función *rvest::html_node()* o *rvest::html_nodes()*.

```
rating <- lego_movie %>%
  html_nodes("strong span") %>%
  html_text() %>%
  as.numeric()
rating
#> [1] 7.7
```

rvest::html_attr()

Esta función extrae el atributo HTML de un nodo que almacene atributos.

Atributos

```
▼ <table class="cast_list">
  ▼ <tbody>
    ▶ <tr>…</tr>
    ▼ <tr class="odd">
      ▼ <td class="primary_photo">
        ▼ <a href="/name/nm0004715/?ref_=tt_cl_i1">
            <img height="44" width="32" alt="Will Arnett" title=
            "Will Arnett" src="https://m.media-amazon.com/images/M/
            MV5BNDkzMjEzNDMyN15BMl5BanBnXkFtZTcwNTk3ODEyOQ@@._V1_UY
            44_CR0,0,32,44_AL_.jpg" class="loadlate"> = $0
          </a>
        </td>
      ▶ <td>…</td>
        <td class="ellipsis"> ... </td>
      ▶ <td class="character">…</td>
      </tr>
```

rvest::html_attr()

Ejemplos

```
cast <- lego_movie %>%
  html_nodes("#titleCast .primary_photo img") %>%
  html_attr("alt")
cast
#>  [1] "Will Arnett"     "Elizabeth Banks" "Craig Berry"     "Alison Brie"
#>  [5] "David Burrows"   "Anthony Daniels" "Charlie Day"     "Amanda Farinos"
#>  [9] "Keith Ferguson"  "Will Ferrell"    "Will Forte"      "Dave Franco"
#> [13] "Morgan Freeman"  "Todd Hansen"     "Jonah Hill"
```

```
poster <- lego_movie %>%
  html_nodes(".poster img") %>%
  html_attr("src")
poster
#> [1] "https://m.media-amazon.com/images/M/MV5BMTg4MDk1ODExN15BMl5BanBnXkFtZTgwNzIyNjg3MDE@._V1_
```

Actividad práctica.

- Repasaremos las funciones vistas previamente y veremos su aplicación en la descarga de datos.