Instituto Tecnológico de Costa Rica

Investigación Web Scrapping

Introducción al Desarrollo Web

Profesor: Efrén Jiménez Delgado

Estudiante: Carlos Fonseca

Grupo 51

Índice

[Introducción 3](#_Toc99395989)

[Web Scrapping aplicado 4](#_Toc99395990)

[Conclusiones 5](#_Toc99395991)

[Recomendaciones 5](#_Toc99395992)

[Bibliografía 6](#_Toc99395993)

# Introducción

El objetivo por cumplir en este proyecto era demostrar lo investigado en el documento anterior al respecto del web scrapping por medio de la aplicación de una de las técnicas, lenguaje y herramientas investigadas.

Haciendo uso del conocimiento adquirido de los XPATH en clase y lo aprendido en la primera parte de esta investigación se logró ejecutar un programa que permite un web scrapping de la información de los vehículos usados en venta en la página de VEINSA MOTORS CR.

# Web Scrapping aplicado

El proyecto se llevo a cabo por medio de la creación de una función de Python que toma manualmente el XPATH de los datos buscados en la página de venta de autos usados de la empresa VEINSA. Los datos obtenidos de los autos por medio de este código serían el modelo, el año del auto, el kilometraje, tipo de transmisión, tipo de combustible, tipo de tracción, el cilindrado, la capacidad de personas, el precio regular y 6 extras del vehículo.

El proceso por el cual se accede a la página es con un entorno virtual dentro del proyecto en el cual se instalaron las librerías de *“requests lxml autopep8”*, *“Beautifulsoup”* y la herramienta *“Selenium”* para poder realizar los procesos por medio de un ChromeDriver.

Por medio del Driver web utilizado por selenium, el programa se dirige a la página en cuestión, entra a cada auto que no haya sido vendido o reservado y revisa el XPATH enviado de cada dato que se desea obtener. Estos datos en un principio son guardados en una lista la cual permite el obtener los datos directamente.

El XPATH de cada uno de los datos buscados sería el siguiente:

1. Modelo: /html/body/form/div/div/div[1]/div[1]/div[1]/h1
2. Año: /html/body/form/div/div/div[1]/div[1]/div[2]/div[1]
3. Kilometraje que tiene al ingresar al sistema: /html/body/form/div/div/div[1]/div[1]/div[2]/div[2]
4. Transmisión del vehículo: /html/body/form/div/div/div[1]/div[1]/div[2]/div[3]
5. Combustible: /html/body/form/div/div/div[1]/div[1]/div[2]/div[4]
6. Tracción: /html/body/form/div/div/div[1]/div[1]/div[2]/div[5]
7. Cilindrado: /html/body/form/div/div/div[1]/div[1]/div[2]/div[6]
8. Capacidad: /html/body/form/div/div/div[1]/div[1]/div[2]/div[7]
9. Precio: /html/body/form/div/div/div[1]/div[1]/div[3]/div[1]
10. Imagen: /html/body/form/div/div/div[1]/div[3]/div/div/div[2]/img
11. Estos serían los extras, los cuales están en una lista dinámica en sucesión y no siempre son los mismos, esta cantidad es la mínima en todas las entradas:
    1. /html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[1]
    2. /html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[2]
    3. /html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[3]
    4. /html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[4]
    5. /html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[5]
    6. /html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[6]

# Conclusiones

Si bien el web scrapping puede ser muy fácil de realizar, se deben tener demasiadas variables en cuenta, como las actualizaciones del sitio web a scrapear que podrían cambiar las direcciones web de las cuales se obtiene la información o los XPATH personalizados de cada dato a extraer.

Adicional a eso está que puede que el sitio web considere como ilegal la obtención de la información que albergan.

# Recomendaciones

Recordar el revisar que los datos de un objeto a otro sigan un estándar entre sí o tengan similitud para evitar que los XPATH de un objeto a otro fallen, además de recordar que por medio del código a veces se tendrá que acceder a una página distinta, en ese salto de página a página el DOM se corromperá, por ende, se debe obtener el enlace y utilizar los datos en otro lado para poder ser utilizados sin problema alguno.

Tomar en cuenta todas las excepciones en los objetos a consultar, por ejemplo, en la página de VEINSA había autos que ya estaban reservados o vendidos y al ejecutar el programa sin tomar en cuenta eso se detendría ya que no encontraría nada al entrar en ese objeto.

Además de recordar que el programa no realiza la paginación de forma automática o el “devolverse” en la página de detalles del objeto puede no ejecutarse y provocar un error en la paginación.

Al usar Selenium, se debe utilizar un Web Driver cuya versión sea acorde al navegador que se utilice puesto que puede no ejecutarse.

Utilizar XPATH’s que sean por clase, id o nombre ya que, si existe una actualización en la página el XPATH actual, como el usado en el proyecto, ya no serviría para futuras ejecuciones.

# Bibliografía

ChromeDriver - WebDriver for Chrome. <https://chromedriver.chromium.org>