INSTITUTO TECNOLÓGICO DE COSTA RICA

INGENIERÍA EN COMPUTACIÓN

Introducción al desarrollo de aplicaciones para web,
Tercera investigación de Web Scrapping

Efrén Jiménez Delgado.

RAFAEL QUESADA ALPÍZAR

SEDE REGIONAL SAN CARLOS

MAYO, 2017

Resumen Ejecutivo

Este documento explica la ejecución de un web scrapper básico en un sitio web para extraer todos sus datos y poder manipularlos de la forma que se desee. Esto se realiza a partir de una investigación anteriormente realizada en la cual se planteaba la estrategia que se siguió en este trabajo, con la variante en el sitio web, ya que en esta etapa se aplica el web Scrapping a la página de Titicupon para los cupones existentes de “Turismo y aventura”. Además este documento explica dicha estrategia planteada anteriormente y su respectiva implementación.

Contenido

Introducción

En la actualidad la capacidad de obtener grandes cantidades de información desde sitios web es muy valiosa ya que hay un sinfín de utilidades que se le podrían dar a esta información, especialmente en ámbitos como el mercadeo y el estudio de tendencias de mercado o incluso brinda posibilidades para nuevas ideas de negocios.

Debido a esto es que ha surgido la necesidad de obtener estas grandes cantidades de información, una tarea que un humano puede hacer sin embargo a grandes cantidades se vuelve muy tedioso, es a este punto donde surge el Web Scrapping basa su funcionamiento en la obtención y estructuración de información de sitios web mediante algoritmos desarrollados especialmente para esto o incluso los conocidos bots.

El Web Scrapping puede ser utilizado de muchas formas, existen bastantes herramientas para su implementación e incluso técnicas especializadas, además de que existe un trasfondo legal el cual es importante de entender así como las herramientas y técnicas anteriormente mencionadas, es por esto que en este documento se abordan estos temas así como un ejemplo de una propuesta para realizar Web Scrapping a la página titicupon.com, específicamente a los cupones de “Turismo y aventura”.

Acerca del Scrapping

Como ya se mencionó anteriormente, para realizar Web Scrapping existen diferentes técnicas las cuales pueden ser:

1. **Obtención manual:**

Esta se refiere a la obtención de los datos por parte de un humano, esta se puede considerar la técnica más lenta, sin embargo es la técnica más minuciosa y permite obtener información que los web scrappers no son capaces de obtener, por ejemplo en casos en los que las páginas de las cuales se desea extraer información implementan técnicas para bloquear la utilización de bots o web scrappers.

Además esta técnica permite aplicar mejores filtros en las páginas para así obtener solo datos estrictamente deseados, sin embargo tiene la gran desventaja que consume una enorme cantidad de recursos, especialmente el tiempo.

2. **Uso de aplicaciones y servicios web:**

Esta técnica se refiere a la utilización de programas especialmente desarrollados y dedicados a la implementación de Web Scrapping, como es de suponer existen algunas de estas herramientas de pago y otras gratis, existen programas de escritorio y servicios web dedicados a esto.

Esta técnica es bastante factible para obtener datos, sin embargo es posible que sean muy generales lo cual no sería de gran ayuda dependiendo de la utilidad que se le desea dar a los datos obtenidos.

Además muchas de estas solo son capaces de obtener datos previamente

estructurados en sus sitios web, esto genera un gran inconveniente para los casos en los cuales se desee recolectar datos de sitios web sin estructuración de datos o desde secciones sin estructurar.

Es por esto que cuando se desee realizar Web Scrapping se debe realizar un análisis a fondo y tomar en cuenta todos estos aspectos. Para brindar un mejor panorama a continuación se muestran algunas de estas herramientas:

- Google Spreadsheet:

  Esta es una herramienta proporcionada dentro de las aplicaciones de Google, esta cuanta con una gran cantidad de funcionalidades como la del uso de fórmulas dentro de las cuales se encuentra “importHTML”, la cual permite al usuario extraer datos de una lista o tabla de algún sitio web.
- Table Capture:

  Esta herramienta es una extensión de Google Chrome, la cual proporciona una funcionalidad similar a la de Google Spreadsheet ya que también permite la obtención de datos desde una tabla o lista de algún sitio web específico.
- Import.io:

  Este servicio web permite obtener información desde otro sitio, tales como imágenes y texto, además puede extraer datos desde mapas. Esta es capaz de generar una API con los datos extraídos.

  Esta es una herramienta que es gratis, sin embargo cuenta con una versión de pago y una versión de escritorio.
- Kimono:

  Esta herramienta cuenta con funcionalidades muy parecidas a Import.io, está también genera una API con los datos extraídos, sin embargo esta cuenta con una característica muy interesante, la cual es que puede obtener datos no estructurados en el sitio web. Esta herramienta también cuenta con una versión gratuita y una de pago.

3. **Programación del Web Scrapper:**

Se refiere a la programación del web scrapper utilizando algún lenguaje de programación especializado, framework o librería. De esta manera se crea un algoritmo de Web Scrapping completamente a la medida, con mayor flexibilidad e integración.

Esta es la técnica más recomendada para hacer Web Scrapping en los casos en los que la recolección es muy compleja, muy específica o muy grande, esto porque se puede programar el web scrapper completamente adaptado a las necesidades que se tengan.

Para llevar a cabo el Web Scrapping con esta técnica se requiere tener conocimientos en el ámbito de la programación, además de una buena implementación de técnicas para evitar que el algoritmo sea bloqueado por los sitios web.

Además existen lenguajes que permiten el parseo de archivos HTML como XQuery y HTQL, además existen multiples lenguajes que cuentan con librería y frameworks para hacer Web Scrapping, por ejemplo Java cuenta con jsoup, C#, Ruby, etc. Sin embargo el mejor candidato parece ser Python, el cual cuenta con una comunidad muy grande y además cuenta con tres librerías muy reconocidas para Web Scrapping, las cuales son Request, BeautifulSoup y Scrapy.

Ámbito legal

Existen muchos proyectos que han sido detenidos o incluso han sido expuestos a grandes demandas por hacer Web Scrapping, sin embargo si se desea hablar de la legalidad del Web Scrapping se debe tener un aspecto muy claro, el cual es el uso que se le dará a la información obtenida, esto porque se dice que si será para uso personal entonces no es ilegal, pero si puede serlo en caso que sea para

comercializar datos o hacer negocio con la información obtenida.

Pese a esto, antes de emprender un proyecto que utilice Web Scrapping, es recomendable informarse acerca de la legalidad que rige en el país del sitio web, sus términos y condiciones de uso.

Metodología de Web Scrapping para Titicupon

Como se mencionó, en este documento se explica una metodología para hacer Web Scrapping al sitio web www.titicupon.com el cual es una página desarrollada para que las personas puedan acceder a cupones los cuales ofrecen ofertas y descuentos en distintos servicios y artículos; para esto se programó el web scrapper con el lenguaje Python, utilizando la librería de Selenium. Se obtuvieron datos de los cupones para poder ser manipulados y posteriormente ser almacenados en una base de datos en Postgres en una siguiente etapa.

Para este sitio se hizo Scrapping de los cupones de “Turismo y aventura” los cuales son publicados en el sitio, para los cuales los usuarios pueden pagar para obtenerlos. Para cada uno de estos cupones se obtuvieron 15 atributos, los cuales se explican y detallan a continuación:

- Título: Titulo del cupon.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[1]/h1
- Subtítulo: Subtitulo del cupón.
    - Xpath: //*[@id=' “Numero de nodo” ']/div[1]/h2
- Imagen: Una imagen del anuncio del cupon.
    - Xpath: //*[@id="views_slideshow_singleframe_div_cupones-block_14_0"]/div/img
- Precio: Precio del cupón.

    - Xpath: //*[@id=’ “Numero de nodo ”’]/div[1]/div[2]/div/h3
- Precio normal: Precio normal sin oferta.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[1]/div[2]/div/h4
- Ahorro: Porcentaje de ahorro.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[1]/div[2]/div/div[1]/span
- Vendidos: Total de cupones vendidos.
    - Xpath: //*[@id=' “Numero de nodo” ']/div[1]/div[2]/div/div[2]/div[1]/p
- Hora de finalización: Horas restantes antes de que termine la oferta.
    - Xpath hora:

      //*[@id=’“Numero de nodo”’]/div[1]/div[2]/div/div[2]/div[2]/div/div[1]/span[1]

    - Xpath minutos:

      //*[@id=’“Numero de nodo”’]/div[1]/div[2]/div/div[2]/div[2]/div/div[2]/span[1]

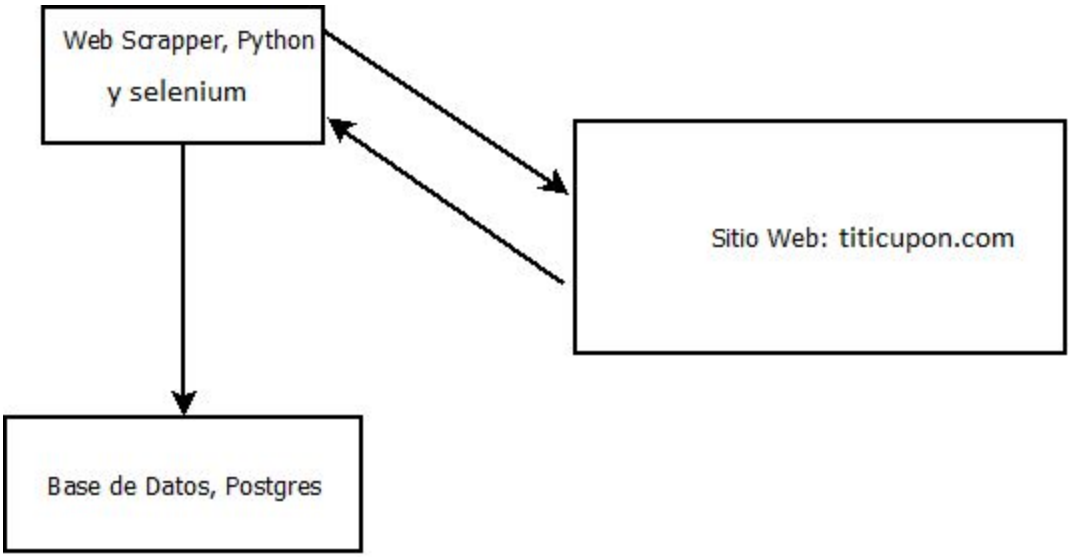
    - Xpath segundos:

      //*[@id=’“Numero de nodo”’]/div[1]/div[2]/div/div[2]/div[2]/div/div[3]/span[1]

- Provincia: Provincia donde se ubica el servicio.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[2]/div[2]/ul/li[1]
- Periodo: Periodo de tiempo de validez del cupon.
    - Xpath: //*[@id=’ ”Numero de nodo” ’]/div[2]/div[2]/ul/li[2]/span[2]
- Horario: Horario en el que puede ser utilizado el cupón.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[2]/div[2]/ul/li[3]
- Dirección: Dirección de la ubicación del servicio.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[2]/div[2]/ul/li[4]

- Ubicación: Ubicación de Google Maps.
    - Xpath: //*[@id="gmap-auto1map-gmap0"]/div/div/div[2]/a
- Sitio web: Enlace al sitio web del servicio.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[3]/div[1]/ul/li[1]/a
- Facebook: Enlace a la página de Facebook.
    - Xpath: //*[@id=’ “Numero de nodo” ’]/div[3]/div[1]/ul/li[2]/a

Para el Xpath el “Numero de nodo” debe ser obtenido para cada cupón y ser insertado en el Xpath de cada elemento que contiene el atributo.



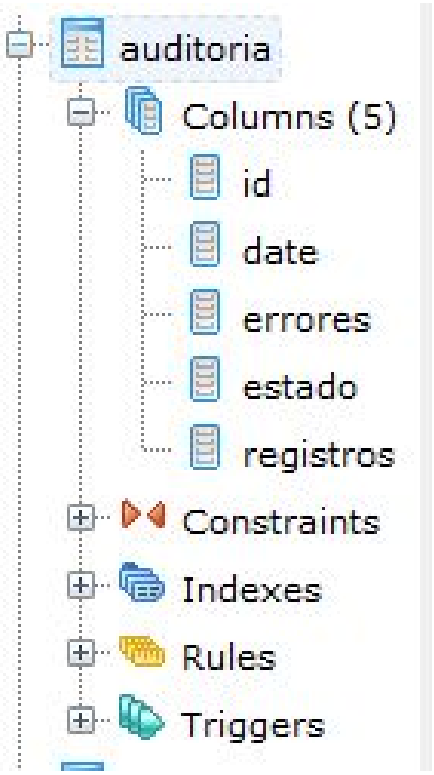
La librería de Selenium es utilizada para abrir el navegador y acceder a cada cupón con su respectivo Xpath, una vez hecho esto se puede obtener la información de cada elemento.

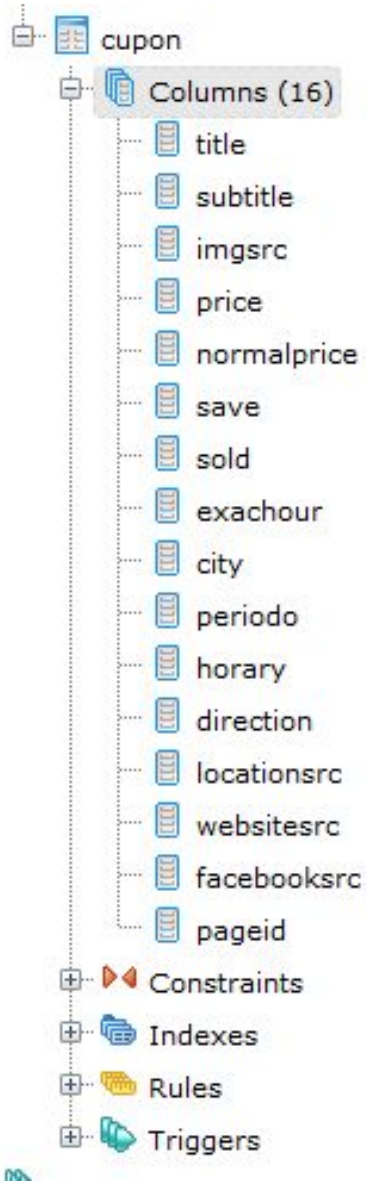
Además como se dijo anteriormente los datos fueron almacenados en una base de datos de Postgres la cual se alojó en Heroku en su versión gratuita. La creación de las tablas e inserción de datos se realizó mediante querys ejecutados desde Python.

Por otra parte se utilizó pgAdmin para verificar e inspeccionar el contenido de la

base de datos utilizada.

A continuación se muestra la estructura de cada una de las tablas insertadas en la base de datos.





De esta forma todos los datos extraídos son almacenados exitosamente y pueden ser accesados configurando un servidor en pgAdmin con los siguientes credenciales:

- database = d2a95d9pumb48g
- user = wykoxseytrhnfs
- password=aa3282757621c9f7d743b071dd82e43aba4ccf9a112d25cd819cea9db3eb0a82

- host = ec2-54-235-90-107.compute-1.amazonaws.com
- port = 5432

Otro aspecto a tomar en consideración es que el código fuente del Scrapper necesita que sea especificada la ruta del ejecutable de Google Chrome para que puede abrir el navegador de forma automática, esto se debe editar en el archivo “scrapper.py” en la línea 20, este archivo se encuentra en el repositorio del proyecto, el cual puede ser consultado en el siguiente enlace: https://github.com/rafaqueal/WebScrapper

Además para ejecutar el scrapper se debe ejecutar el archivo mencionado con el nombre de “scrapper.py”.

Conclusiones y recomendaciones

Para concluir se puede decir que el Web Scrapping puede ser muy útil dependiendo a las necesidades y tiene una gran cantidad de posibles implementaciones y un gran mercado por delante, por esto desarrollar un proyecto con esta metodología puede llegar a ser muy factible.

Además para utilizar el Web Scrapping existen diferentes técnicas y herramientas que se ajustan según las necesidades que se tengan, por este motivo es necesario realizar un análisis detallado para así comprobar cual técnica o herramienta es la más adecuada en caso de emprender un proyecto con estas características.

Por otra parte es muy importante tener en cuenta las implicaciones legales que se pueden tener, por este motivo se recomienda informarse bien sobre la legalidad en vigencia y los términos de uso de los sitios a los cuales se les deseen hacer Web Scrapping.

Bibliografía

BBVAOPEN4U. (1 de enero de 2016). *BBVAOPEN4U*. Obtenido de Herramientas de extracción de datos: para principiantes y profesionales: https://bbvaopen4u.com/es/actualidad/herramientas-de-extraccion-de-datos-para-principiantes-y-profesionales

FelicianoBV. (4 de noviembre de 2016). *Feliciano Borrego*. Obtenido de Alternativas para realizar web scraping: http://felicianoborrego.com/alternativas-para-realizar-web-scraping/

Marq, M. (8 de abril de 2016). *sitelabs*. Obtenido de Qué es el Web scraping? Introducción y herramientas: https://sitelabs.es/web-scraping-introduccion-y-herramientas/

Moody, D. (s.f.). *What is the best programming language for web scraping?* Obtenido de hexfox: https://hexfox.com/p/what-is-the-best-language-for-web-scraping/

Pawlas, P. D. (2012). Universal web pages content parser. *Communications in Computer and Information Science*, 130-138.

Sanjay Kumar Malik, S. R. (2011). Information Extraction Using Web Usage Mining, Web Scrapping and Semantic Annotation. *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*.