

Instituto Tecnológico de Costa Rica

Ingeniería en Computación

Tarea de Investigación: Web Scraping Tercera Parte

Keylor Jesús Guevara Jiménez

Sede San Carlos

15 – 05 – 17

Resumen Ejecutivo

Dentro de este tercer documento se estarán analizando los diferentes procesos que son necesarios realizar para poder alcanzar con éxito la ejecución de la metodología descrita en el entregable número uno y números dos de esta misma investigación sobre como poder realizar un proceso de web scraping adecuado de una página de compra y muestreo de libros. Podremos realizar una conexión directa con una base de datos para almacenar la información obtenida de nuestra página web.

Largas han sido las horas de desarrollo en las que con gran orgullo el éxito es alcanzado, el poder desarrollar este tipo de procedimientos tan avanzados nos demuestran que el avance dentro de la educación universitaria no esta tan lejos, o al menos desde un pequeño ángulo, de la realidad laboral que nos espera.

Contenido

Resumen Ejecutivo	2
Introducción.....	4
Desarrollo	6
Conclusiones y Recomendaciones	12
Conclusiones.....	13
Recomendaciones	13
Bibliografía.....	14

Introducción

Tomando de referencia el primer documento de esta investigación recordemos que web scraping es un término que no surgió hace poco, es tan antiguo como el mismo internet (Mitchell, 2015) solo que durante muchos años su nombre ha ido cambiando pasando por “screen scraping”, “data mining”, “web harvesting” y por otras variantes similares, solo que su nombre se ha mantenido de una manera mucho más estable bajo el llamado a web scraping.

Este término [“web scraping”] trata de envolver todo el proceso correspondiente a recopilar datos de una dirección web a través de cualquier otra forma que no sea por medio de un programa que interactúa de una forma directa con una API (Mitchell, 2015). Obteniendo todos estos datos, y teniendo la posibilidad de ser guardados en una base de datos, en la cual, estos mismos de ser usados de cualquier forma por parte de su administrador (Kumar Mahto & Singh, 2016).

Recordando que la página definida para el desarrollo de extracción corresponde a <http://www.e-libro.com/> una página dedicada a ofrecer textos completos, desde textos de cátedra, libros, hasta artículos, pasando por investigaciones científicas de todas las disciplinas académicas.

Dentro del proceso de selección se tomaron en cuenta diferentes puntos para poder así hacer la mejor selección, sin duda el contar con un ítem capaz de contener la mayor cantidad de atributos era uno de los pasos claves, y sin duda la página e-libro cumplía con el requisito ya que al ser un ofertante de libros traía atributos que son identificables siempre en un libro, su autor, su editorial y su precio, así como datos más específicos solicitados por el profesor como el poder obtener una imagen o el tener datos en formato de fecha, todo esto, logrado en esta página, ya que los libros cuentan con una portada así como con una fecha de publicación.

Realizando un proceso de vuelta atrás encontramos necesario recordar cuales son algunos de los elementos que deben de ser considerados como importantes de mencionar como lo son el lenguaje de programación y sus librerías a utilizar:

Lenguaje de programación	Python
Librerías utilizadas	
Urllib	Pymysql
urllib2	BeautifulSoup4
Re	Lxml
Json	Sys

Desarrollo

Dentro del desarrollo que se da dentro del proyecto llegamos a la etapa de realizar la conexión con la base de datos; la cual se encuentra compuesta por dos tablas que representan el manejo principal de los datos, la tabla de registros y la tabla de auditoria, en las cuales, se almacena la información con respecto a todos los atributos obtenidos del proceso de web scraping y la otra lleva un control del proceso de web scraping, donde se podrá observar la cantidad de procesos realizados y diferentes conceptos regulares de una auditoria.

El proceso de conexión de la información que se tiene mediante el scrapeo de los datos y la base de datos es sumamente dinámico y nos muestra la capacidad que tiene en este caso herramientas como lo son BS4 en Python, para manejar de una forma veloz y ordenada los datos de las páginas victimas.

Las tablas desarrolladas a continuación representan los diferentes datos que se encuentran almacenados dentro de la base de datos. Dichas tablas están representadas a continuación:

Tabla Registros	
Nombre Columna	Tipo Asignado
idRegistro	Int
Titulo	Varchar
nombreAutor	Varchar
nombreEditorial	Varchar
fechaPublicacion	Varchar
portadaLibro	Varchar
precioExclusivo	Varchar
threeUser	Varchar
precioGeneral	Varchar

Tabla Auditoria	
Nombre Columna	Tipo Asignado
idAuditoria	Int
paginaWeb	Varchar
Fecha	Varchar
Error	Varchar
Estado	Varchar

Para tener una mejor visión de estos elementos en el entregable de Investigación dos se presenta la siguiente tabla que contiene los atributos que son obtenidos del proceso de web scraping de la página e-libros:

Nombre	Descripción
Cover_book	Es la imagen de la portada del libro
Title	El título del libro
Autor	Incluye el nombre del autor del libro
Editorial	Incluye el nombre de la editorial a la que pertenece el libro.
Temas	Incluye los temas a los que se encuentra asociado el libro
Identificación del documento	Incluye información acerca de datos correspondientes al libro
pISBN	Número de rastreo de los libros digitales
eISBN	Número de identificación del libro.
Precio de acceso exclusivo	Ofrece un precio único que hay para el libro.
Three user	Precio disponible para tres usuarios
Precio para múltiples usuarios	Precios que tienen múltiples usuarios al querer adquirir el libro.

El proceso por el cual se logra obtener toda la información dentro de las páginas de la librería digital es explicado a continuación:

1. Se define la conexión de la base de datos que se va a implementar, recordar que el lenguaje seleccionado para realizar el proyecto fue Python:
 - a. Mostramos el string de conexión necesario para conectar las dos partes del proyecto:

```
# Coneccion con la base de datos MySQL
c = pymysql.connect(host="webscraping.cwreudtoe38d.us-west-
2.rds.amazonaws.com", user="masterUser", db="webscraping",
passwd="webscraping", use_unicode=True, charset="utf8")
```

2. Luego de conectar el proyecto del lenguaje Python junto a una base de datos que en este caso corresponde a MySQL, procedemos a declarar una instancia para el uso de la librería especializada en web scraping, la cual corresponde a:

```
bsObj = BeautifulSoup(html, "lxml") # objeto de la clase bs4
```

3. Una vez diseñado esto procedemos a recorrer todo el objeto y obteniendo los datos que sean deseados para el poder luego tomarlos y guardarlos en la base de datos.

Se muestra a continuación la forma en que se debe de realizar la toma de algún valor, por ejemplo, el título del libro:

```
titulo = child.find("div", {"class": "book_info_titlelist"}).find("a", {"class": "title"}) #
TITULO_LIBRO
```

4. Una vez tomada la información procedemos a almacenar la información en la base de datos de la siguiente forma:

```
try:
    with c.cursor() as cursor:
```



```

sql = "INSERT INTO `registros`
(`titulo`,`nombreAutor`,`nombreEditorial`,`fechaPublicacion`,`\
`temaTratado`,`identificacionDocumento`,`pISBN`,`eISBN`,`precioExclusivo`,`threeUser`,`\
`multiplesUsuarios`,`imagenPortada`)
VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s) "
cursor.execute(sql, [
    tituloString, nombreAutorString, nombreEditorialString,
fechaPublicacionString, temaTratadoString,
    identificacionDocumentoString, pISBNString, eISBNString,
precioExclusivoString, threeUserString,
    usuariosMultiplesString, imagenString])
c.commit()
finally:
    pass

```

5. Una vez realizado dichos pasos en Python, nada más procede a visualizarlos en la base de datos.

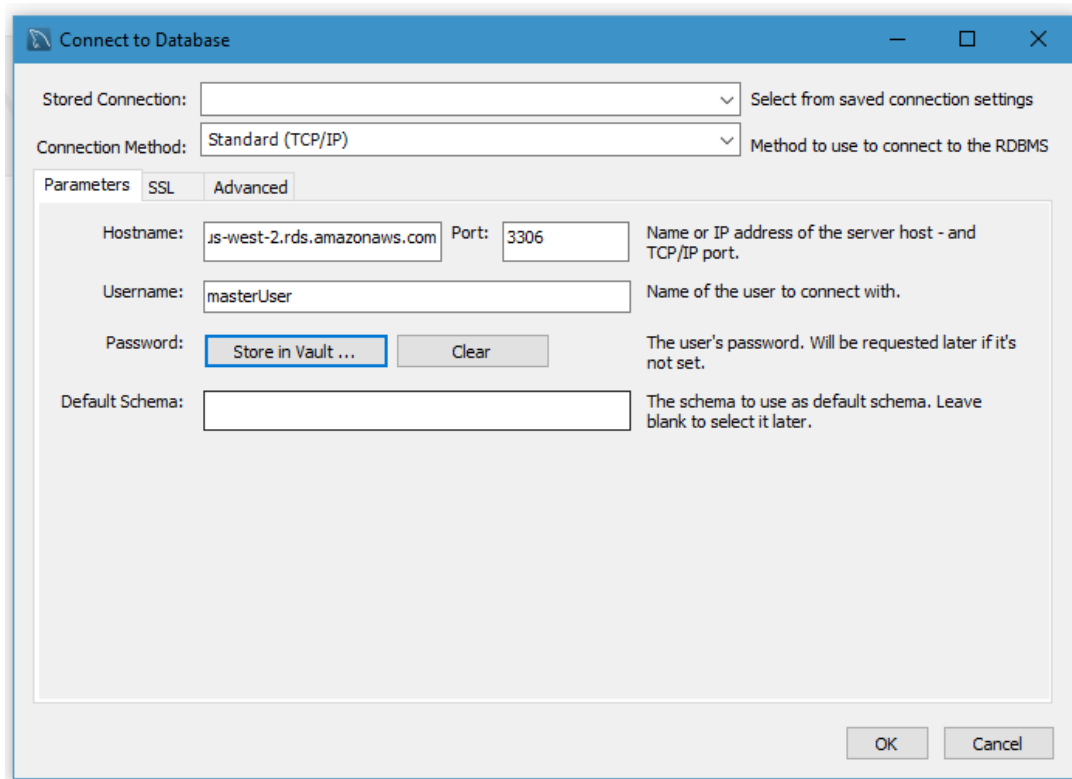
- a. Para lograr la conexión con la base de datos se realizó un proceso de prueba en el que se realizó la inserción dentro de una de las bases de RDS proveída por el servicio de Amazon.

Mediante la cual nos da posibilidad de poder realizar las consultas o inserciones sin necesidad de tener que implementar una base de datos en nuestro localhost, sino que basta con implementar unas cuantas credenciales dentro de algún Administrador de bases de datos MySQL. **Necesita de acceso a Internet.**

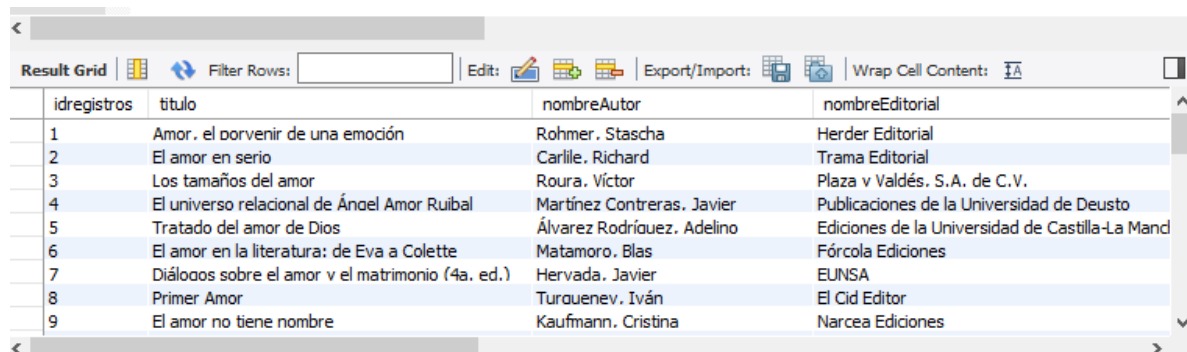
Credenciales para poder visualizar la base de datos: Para realizar el logueo a la base de datos que Amazon dispone para nuestros datos basta con ingresar los siguientes datos en algún administrador de

bases de MySQL, para esta investigación se desarrolla mediante MySQL Workbench:

- Hostname: webscraping2017.cwreudtoe38d.us-west-2.rds.amazonaws.com Port: 3306
- Username: masterUser Password: webscraping2017



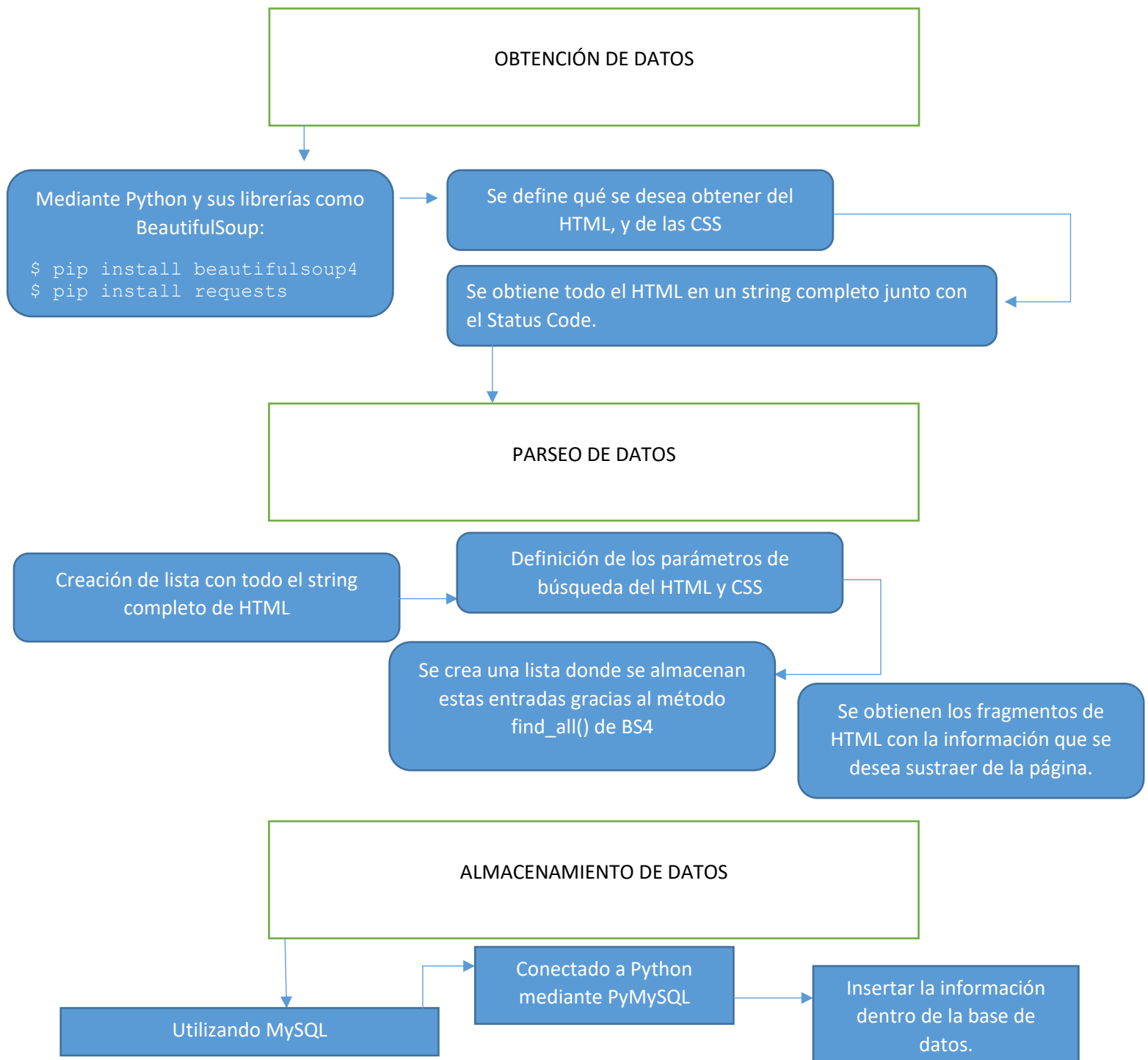
6. Luego de esto podemos proceder a ver los datos que nuestra base RDS contiene y va obteniendo mediante el web scraping desarrollado en Python.



idregistros	titulo	nombreAutor	nombreEditorial
1	Amor, el porvenir de una emoción	Rohmer, Stascha	Herder Editorial
2	El amor en serio	Carlile, Richard	Trama Editorial
3	Los tamaños del amor	Roura, Víctor	Plaza v Valdés, S.A. de C.V.
4	El universo relacional de Ángel Amor Ruibal	Martínez Contreras, Javier	Publicaciones de la Universidad de Deusto
5	Tratado del amor de Dios	Álvarez Rodríguez, Adelino	Ediciones de la Universidad de Castilla-La Mand
6	El amor en la literatura: de Eva a Colette	Matamoro, Blas	Fórcola Ediciones
7	Diálogos sobre el amor v el matrimonio (4a. ed.)	Hervada, Javier	EUNSA
8	Primer Amor	Turquenev, Iván	El Cid Editor
9	El amor no tiene nombre	Kaufmann, Cristina	Narcea Ediciones

Como último paso en vez de implementar EC2 o Azure, se decide almacenar en Heroku, para poder tener un manejo global del proyecto.

Recordemos el modelo definido inicialmente en el primer segmento de este documento la metodología que se iba a seguir mediante el siguiente diagrama.



Links

Github:

<https://github.com/KeylorGuevara/webScraping3InvestigacionPython.git>

Heroku:

<https://webscraping2017.herokuapp.com/>

Conclusiones y Recomendaciones

Conclusiones

- En la actualidad existen gran variedad de métodos para realizar un proceso de “web scraping”, cada uno tiene ventajas y desventajas sobre ellos mismos, lo importante es escoger el mejor método según las necesidades de extracción que se tienen.
- Herramientas para desarrollar “web scraping” existen en gran variedad, cada una cuenta con sus ventajas y sus desventajas todo depende del conocimiento del usuario que la implemente así como la utilidad con la que use esta herramienta.
- Indagar páginas y comunidades confiables, como la misma comunidad de Python la cual cuenta con gran cantidad de datos importantes sobre librerías y demás herramientas sumamente eficientes según nuestras necesidades.

Recomendaciones

- Analizar adecuadamente cuál método escoger para realizar “web scraping” para de esta forma evitar la pérdida de tiempo y de recursos económicos por parte del usuario.
- Controlar de una forma adecuada todas las diferentes versiones de código que estemos implementando en nuestro trabajo.
- Diseñar diferentes estrategias para poder identificar cuáles son los segmentos de HTML que realmente estamos tratando de identificar, por ejemplo, existen extensiones dentro de ciertos buscadores diseñadas específicamente para identificar las partes del HTML.

Bibliografía

- Borrego, F. (9 de Marzo de 2016). Alternativas para realizar web scraping. Obtenido de Feliciano Borrego: <http://felicianoborrego.com/alternativas-para-realizar-web-scraping/>
- Crummy.com. (2004). Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation. [online] Disponible desde: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> [Accesado 1 Abr. 2017].
- FEDERAL COURT OF AUSTRALIA. (08 de Marzo de 2017). Federal Court of Australia. Obtenido de Federal Court of Australia: <http://www.austlii.edu.au/au/cases/cth/FCA/2010/44.html>
- Kumar Mahto, D., & Singh, L. (2016). A Dive into Web Scraper World (Vol. 3rd International Conference). Delhi, India: IEEE.
- Lawson, R. (2015). Web Scraping with Python. Packt Publishing Ltd.
- Martí, M. (09 de 03 de 2017). sitelabs. Obtenido de Qué es el Web scraping? Introducción y herramientas: <https://sitelabs.es/web-scraping-introduccion-y-herramientas/>
- Mitchell, R. (2015). Web Scraping with Python: *Collecting Data from the Modern Web*. California: O'Reilly Media, Inc.
- Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2014). Automated Data Collection with R – A Practical Guide to Web Scraping and Text Mining. En S. Munzert, C. Rubba, P. Meißner, & D. Nyhuis, *Journal of Statistical Software* (pág. 480). Chichester: John Wiley & Sons.