Web Scraping de los Perfiles y Publicaciones de una Afiliación en Google Scholar utilizando Aplicaciones Web e implementando un Algoritmo en R

Web Scraping of the Profiles and Publications of an Affiliation in Google Scholar using Web Applications and implementing an Algorithm in R

Danny Murillo [1], Dalys Saavedra [2]
[1,2] VIPE, Universidad Tecnológica de Panamá
[1] danny.murillo@utp.ac.pa,[2] dalys.saavedra@utp.ac.pa

Resumen– *El objetivo de este artículo es hacer uso de la técnica Web Scraping para extraer datos de Google Scholar a través de diferentes métodos. El Web Scraping es una forma de minería de datos no estructurada, que permite extraer información de páginas web, escanear su código HTML y generar patrones de extracción de datos. El artículo muestra las pruebas realizadas de estos métodos para medir la velocidad de extracción de los datos y buscar la mejor forma de extraer los datos de GS de forma estructurada. El artículo también muestra el análisis y desarrollo de un algoritmo en el lenguaje R, para comparar la velocidad de extracción de los datos y la eficiencia en el formato de salida de los datos.*

Palabras claves– *Web Scraping, Google Scholar, Minería de datos, Lenguaje R, análisis de datos.*

Abstract– *The objective of this article is to make use of the Web Scraping technique to extract data from Google Scholar through different methods. Web Scraping is a form of unstructured data mining, which allows you to extract information from web pages, scan your HTML code and generate data extraction patterns. The article shows the tests performed by these methods to measure the speed of extraction of the data and to find the best way to extract the GS data in a structured way. The article also shows the analysis and development of an algorithm in the R language, performing tests to 15 profiles of universities in Google Scholar, comparing with other methods of scraping, data extraction speed and efficiency in data output format.*

Keywords– *Web Scraping, Google Scholar, Text Mining, R Lenguage, data analysis.*

1. Introducción

Internet, una red de redes que se construye a partir de 1969 y que aún al inicio de 1980 era básicamente una red física de redes, máquinas y cables interconectados que permitían enviar paquetes de información entre computadoras, según Tim Berner-Lee, la idea de la web era diseñar un espacio de trabajo colaborativo que facilitará el flujo de información [1]. Realmente, tal y como la gente lo entiende ahora fue en 1994, a partir de la existencia de un navegador web que se integra con la World Wide Web [2] y la vinculación de páginas con código html, enlaces de hipertexto, contenido multimedia, la WWW dejó de ser una red de enlaces entre páginas y documentos evolucionando a una red de datos [3].

Un gran número de expertos considera que el principal defecto del actual modelo de Internet radica en esa sobreabundancia de información, cuyo tratamiento exige una enorme cantidad de tiempo y energía a fin de cribar la calidad de los datos sumergidos en tan enorme repositorio de datos [4]. Estos datos están organizados, estructurados y visibles en páginas web, pero, no siempre es posible poder extraerlos, reutilizarlos o analizarlos con la rapidez o la estructura deseada. .
Uno de estos ejemplos, es la web de Google Scholar (GS), un buscador de Google lanzado en noviembre de 2004, enfocado al ámbito académico donde se almacena un extenso conjunto de trabajos de investigación científica incluyendo los de acceso abierto[5]. GS

recopila la producción científica de un investigador y la ofrece agregada en una página web, añadiendo información sobre el número de citas de cada referencia [6] que proviene de publicaciones realizadas en conferencias, congresos. Es un producto que, a diferencia de las bases de datos bibliográficas tradicionales, no vacía contenidos de revistas, sino que rastrea sistemáticamente la Web siguiendo la misma filosofía que Google, pero haciendo converger en una sola plataforma diferentes servicios. [7]
Los datos de GS son relevantes para realizar análisis Bibliométrico de los perfiles, revistas indexadas como las citaciones de cada uno de estos perfiles y publicaciones a nivel mundial, sin embargo, para poder obtener estos datos es necesario utilizar alguna técnica de Minería de texto debido a que GS no cuenta con ninguna forma para extraer sus contenidos. La minería de texto es una forma de extraer información de un conjunto de datos, ésta, integra la minería de contenido web, que contiene 4 formas de extracción: minería de datos no estructurada, minería de datos estructurada, minería de datos semi-estructurada, extracción de datos multimedia [8]. En la "Minería de datos no estructurada" está la minería de páginas web, que utiliza la técnica de web scraping [9].

El "Web Scraping" es una técnica que consiste en la extracción de una o varias páginas web de un sitio web que estén relacionadas mediante enlaces, para su manipulación, procesar parte de su contenido y análisis posterior de los datos [10]. Para hacer Web scraping es necesario analizar aspectos cómo: Accesibilidad de los datos de origen, análisis de patrones de los datos, frecuencia de extracción de los datos con el objetivo de buscar la vía más óptima para obtener los datos.

Este trabajo muestra la evaluación de varias herramientas de scraping para extraer datos de los perfiles y publicaciones de GS, llegando a la conclusión por el tiempo de procesamiento y la no estructuración adecuada en el momento de la extracción, implementar un algoritmo utilizando el lenguaje R. R es un lenguaje de programación de código abierto, desarrollado por el grupo Core Team [11]. Es un lenguaje de script por lo que no requiere ser compilado para ser ejecutado y tiene similitud con otros lenguajes como C o C++, mezcla diferentes características de otros lenguajes.

Trabajos previos de funciones en lenguaje R para extracción de datos en Google Scholar

Función en R "GScholarScraper" : es una función en R creada en el 2012 por Kay Cichini, permite Scrapear los perfiles y detalles de las publicaciones de un perfil en Google Scholar, pero, solo permite extraer un perfil a la vez y no muestra a que perfil pertenecen las publicaciones extraídas, ni a que afiliación. Su última actualización fue en noviembre de 2016 [12].

Paquete en R llamado "scholar": es un paquete en R que proporciona funciones para extraer datos de GS. Fue creado por James Keirsted en el 2015 y su última actualización es de junio de 2016. Se utilizó la función get_profile() para extraer el perfil por separado y la función get_publications() para extraer los detalles de las publicaciones, pero, no indica a que usuario de GS pertenece los detalles de las publicaciones extraída [13].

2. Antecedentes

Antes de crear el algoritmo en R se realizó una evaluación de varios métodos de web scraping para comparar y evaluar la velocidad de extracción de cada método y la estructura de salida al extraer los datos.

2.1 Selección de Métodos

Realizamos las pruebas utilizando 4 métodos: copiar y pegar, Local Browser, Local Software, Online "Tabla 1. Ha excepción del método de copiar y pegar, fue posible exportar los datos en formato .CSV, no sin antes realizar un proceso de depuración de los datos debido a que los datos que se extraen están unidos a otros textos que no eran de interés.

Tabla 1. Método de web scraping, aplicación a utilizar y facilidad de uso del método.

Métodos	Aplicación	Descarga	Conocimientos del usuario	Facilidad de uso
Copiar y Pegar	Manual	ninguno	ninguno	Fácil
Web scraping Local Browser	Extensión Chrome	Gratuito	Técnico mínimo	Fácil
Web scraping Local Software	Fminer	Pago (Trial)	Técnico intermedio	No es Fácil
Web Scraping Online	Import.io	Pago (Free versión)	Técnico mínimo	Fácil

2.2 Aplicaciones utilizadas para cada método

Copiar y Pegar: no es un método de Web Scraping, pero es la forma más común de extraer datos de un sitio web, el proceso consistió en copiar y pegar cada dato del perfil y las publicaciones en una tabla de Excel, seleccionando solo el dato que se necesitaban, pero el trabajo resulto muy extenso.

Web scraping Local: se utilizó la extensión SCRAPER de Google Chrome. Permite seleccionar un bloque de datos de una página web y al activar la extensión, extrae los datos que tengan el mismo patrón de la clase HTML seleccionada, Solo permite scrapear los datos una página por vez del perfil de Afiliación, por lo que el ciclo de repetición de Web Scraping lo debe hacer el usuario [14].

Web scraping Local Software: se utilizó el software FMiner, al abrir la página web en la aplicación, permite grabar el proceso como un macro donde se va creando un diagrama de flujo de datos de la página web asignando el valor seleccionado a cada variable, el proceso es semi-automático, ya que el usuario debe escoger cuales son los datos que nos interesa guardar [15].

Web scraping Online: Import.io es una aplicación Online que analiza automáticamente la estructura de la página web y muestra los datos en formato de tabla, es posible extraer datos de paginación, sin embargo, en las pruebas realizadas no lograba identificar las páginas siguientes, por lo que aumentaba el tiempo de extracción de los datos [16].

2.3 Selección de datos

Para realizar las pruebas se seleccionaron 5 perfiles de Universidades en GS: Universidad Fransisco Marroquin (UFM), Escuela Superior Politécnica del Litoral (ESPOL), Universidade Regional de Blumenau (FURB), Universidad Tecnológica de Panamá (UTP), Universidad de La Habana (UH). Para cada Universidad se contabilizó el número de perfiles y publicaciones que tenía cada una en GS "Tabla 2".

Tabla 2. Perfiles de universidades seleccionadas para web scraping en google scholar.

Universidad	País	#Perfiles	#Publicaciones
UFM	Guatemala	14	393
ESPOL	Ecuador	67	1061
FURB	Brasil	38	1360
UTP	Panamá	77	1434
UH	Cuba	79	2758

2.4 Pruebas de los métodos Web de Scraping

Para cada perfil de las Universidades seleccionadas se aplicó cada uno de los métodos seleccionados donde se extrajeron todos los perfiles y las publicaciones de cada perfil, donde se midió el tiempo de extracción en minutos.

El resultado de estas pruebas muestra que el **método de Web Scaping Online** obtuvo el mejor tiempo promedio de extracción de datos de una Universidad con 35 perfiles y 466 publicaciones, el cual fue de **2 horas 18 minutos** "Tabla 3". En estas pruebas no se consideró extraer los detalles de cada publicación, por lo que el tiempo pudo ser mayor en cada método.

Tabla 3. Resultado del tiempo promedio de scraper por método, de los perfiles y publicaciones de las 5 universidades en gs.

	TIEMPO POR MÉTODO WEB SCRAPING Perfiles / Publicaciones (minutos)			
Universidad	**Copiar / Pegar**	**Local Browser**	**Local Software**	**Online**
UFM	8 / 130	2 / 35	3 / 50	2 / 35
ESPOL	42 / 354	9 / 95	14 / 140	9 / 94
FURB	24 / 445	5 / 122	8 / 179	5 / 120
UTP	50 / 482	11/129	17 / 189	10 / 127
UH	51 / 920	11 / 245	17 / 363	10 / 244
Promedio	**35 / 466**	**8 / 125**	**12 / 184**	**7 / 124**

Aunque es posible ver la disminución de los tiempos de scraper entre un método y otro, decidimos realizar esta prueba creando un algoritmo en el lenguaje R para hacer un proceso automatizado de extracción, extraer los datos personalizados, disminuir el tiempo de extracción y extraer los datos de forma estructurada.

3. Metodología

3.1 Recursos

- R commander
- Aplicación R studio para Windows
- Paquete en R (rvest) para leer todo el contenido HTML de una página web (web scraping).
- Paquetes en R: xml2, plyr, wordcloud, dplyr, plot, ggplot2.
- Computador con Windows 7 de 64 Bits, Dual Core de 2.2 GHz, y Memoria RAM de 3 GB.
- La velocidad de Internet en periodo de pruebas fue de 1.45 Mb de descarga y 1.90 de Carga.
- Datos de afiliación de 15 Universidades en GS.

3.2. Análisis de estructura de Datos

Se Analizaron los datos de cada bloque de los perfiles identificando patrones repetitivos en los códigos, en la "Figura 1" se muestra los fragmentos de cada perfil que son similares en cuanto al contenido y estructura por lo que resultó más fácil crear ciclos de repetición para extraer esto datos [14].

Figura 1. Listado de Perfiles de Google Scholar, afiliación Universidad Tecnológica de Panamá.

Para poder realizar la extracción es necesario conocer el código html que compone cada bloque de perfil en GS. En la "Figura. 2" se muestra el código HTML extraído del perfil GS, donde se puede ver las etiquetas en rojo que encierran los datos que nos interesan de este perfil (negritas).

```
<div class="gsc_1usr gs_scl">
    <div class="gsc_1usr_photo">
        <a href="/citations?user=l8gpxI4AAAAJ&amp;hl=es"><img
        src="/citations?view_op=view_photo&amp;user=l8gpxI4A
        AAAJ&amp;citpid=1" sizes="(max-width:599px) 75px,(max-
        width:1251px) 100px, 120px"
        srcset="/citations?view_op=view_photo&amp;user=l8gpxI
        4AAAAJ&amp;citpid=1
        150w,/citations?view_op=medium_photo&amp;user=l8gp
        xI4AAAAJ&amp;citpid=1 300w" alt="Elida de
        Obaldia"></a>
    </div>
    <div class="gsc_1usr_text">
        <h3 class="gsc_1usr_name">
            <a
            href="/citations?user=l8gpxI4AAAAJ&amp;hl=es">
            Elida de Obaldia</a>
        </h3>
    <div class="gsc_1usr_aff">Universidad Tecnologica de
Panama</div>
```

Figura 2. Estructura HTML de bloque de perfil Scrapeado.

3.3. Búsqueda de patrones

Evaluamos el código HTML extraído de cada bloque de perfil en GS para buscar si los códigos html que contienen los datos tienen el mismo patrón y esquema de datos "Figura 3", Separamos cada uno de los nodos HTML que contenían los datos de los elementos individuales que serán almacenados en variables para

luego agruparlas en una tabla en R llamada data.frame, esta permite almacenar diferentes tipos de datos.

Variable	Valor HTML	Resultado
url_perfil	read_html(url_GS)	Código HTML completo de primera página
Afiliación	html_text(,url_perfil, h2.gsc_authors_header)	Universidad Tecnológica
Perfil	html_node(url_perfil, div.gs_scl)	Código HML de perfil GS
Nombre	html_text(Perfil, h3>a)	Elida Obaldía
Url_perfil	html_atttr(Perfil, href)	https://scholar.google.es/citations?user=I8gpxI4AAAAJ&hl=es
Id_user	extraer_cadena(Url_perfil)	I8gpxI4AAAAJ

Figura 3. Scraper de datos por valor HTML, los resultados que se obtienen de cada bloque y las variables asignadas.

Analizamos cada bloque extraído de los perfiles de la primera página de GS, en ella se muestra una clase CSS que enmarca el contenido de cada perfil, esta clase *div.gs_scl es* un nodo que se repite, al utilizar la función de Scraper en R html_nodes(url_afiliacion,"div.gs_scl") con el parámetros de la clase identificada, R mostraró los bloques de contenido extraído que cumplían con este patrón dentro del código, que en total deben ser 10 nodos de perfil por cada página.

3.4 Análisis de los datos de los perfiles de GS

Cada afiliación en GS está compuesta por el listado de perfiles con si ID_USER, cada perfil contiene el listado de publicaciones y cada publicación tiene sus detalles, por lo que en el algoritmo que desarrollamos se realizó esta estructura de forma dinámica utilizando dos procesos separados para extraer primero los perfiles y su ID y luego los detalles de las publicaciones.

3.5 Esquema de los algoritmos en R

3.5.1. Algorimo para extraer los perfiles en GS

Se desarrolló un algoritmo para extraer todos los ID de los perfiles de una afiliación utilizando la URL de afiliación de una Universidad en GS. Este algoritmo extrajo el enlace de cada perfil incluyendo los datos de: Nombre, afiliación, palabras claves, citaciones, citaciones 2011, hindex, hindex_2011 y vincular la URL del perfil y el ID del perfil "Figura 4", para luego almacenarlos de manera temporal hasta que terminara el ciclo de repetición, al finalizar los datos se guardaron en un conjunto de datos (data.frame) que se podía accesar y visualizar al terminar el Scraper de la afiliación.

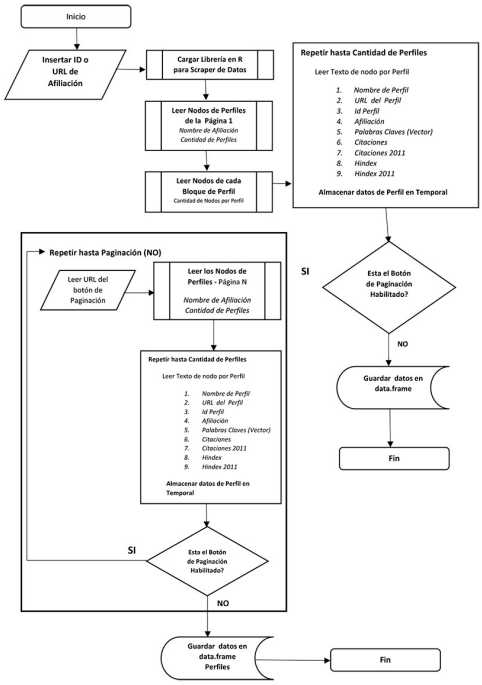

Figura 4. Esquema de Algoritmo para Scrapear datos de Perfiles de una Afiliación en Google Scholar.

3.3.5.2 Algorimo para extraer publicaciones en GS

Se creó un segundo algoritmo para extraer todas las publicaciones por perfil, en este algoritmo se utilizó el paquete "scholar" de R y la función get_publications() que permitió extraer las publicaciones y los detalles de cada. El algoritmo utilizó la tabla creada en el algoritmo 1 para contabilizar el número de perfiles a extraer, y utilizar las URL y nombres de cada perfil de la tabla para añadirlo a las publicaciones extraídas "Figura 5". El número de columnas de los detalles de las publicaciones fue dinámica debido a que algunas publicaciones tenían un esquema de datos de revistas, congresos, libros, estos datos también fueron almacenados en un conjunto de datos.

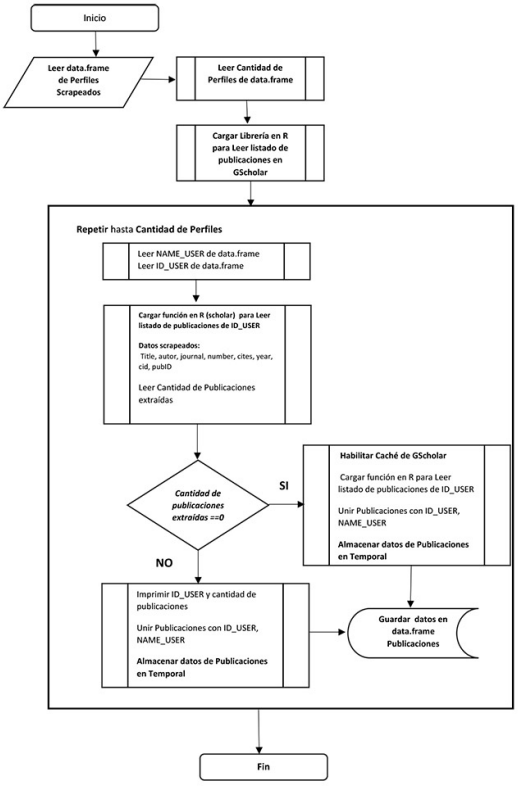

Figura 5. Esquema de Algoritmo para Scrapear todas las Publicaciones por perfil de una Afiliación en Google Scholar.

4. Resultados

4.1. Comparación de métodos y algoritmo en R

Se realizó una evaluación del Algoritmo en R utilizando los datos de las 5 Universidades anteriores con 55 perfiles y 1400 publicaciones. Las pruebas utilizando el algoritmo en R, método "1 algoritmo en R" es de 3 minutos incluyendo perfiles, publicaciones y detalles de las publicaciones "Tabla 4". El algoritmo generó los datos extraídos de forma estructurada en R, que fueron exportados a .CSV y a MS Excel. "Figura 6".

	Nombre	citas	word_key	url_user	id_user	citaciones	cita_2011	hindex	hindex_2011	index10	index10_2011
1	Reinhardt Pinzon Adames	935		/citations?user=1TICvwUAAAAJ&hl=es&oe=ASCII	1TICvwUAAAAJ	935	452	10	6	10	5
2	Elida de Obaldia	880	Material Science, Renewable Energy, Diamond thin fil...	/citations?user=IBgqcr4AAAAJ&hl=es&oe=ASCII	IBgqcr4AAAAJ	880	272	9	5	9	1
3	Maritza Morales Batista	467	inteligencia artificial, open data, linked data, big data,...	/citations?user=Udc5suYAAAAJ&hl=es&oe=ASCII	Udc5suYAAAAJ	467	210	11	8	11	5
4	Oscar M. Ramirez	429	Structural and Earthquake Engineering	/citations?user=Ka8MsMAAAAJ&hl=es&oe=ASCII	Ka8MsMAAAAJ	429	259	6	6	8	5
5	Rodney Delgado-Serrano	413	Astrophysics, Astronomy, Physics, Mathematics, Info...	/citations?user=BUF9cMAAAAJ&hl=es&oe=ASCII	BUF9cMAAAAJ	413	234	6	6	6	6
6	Gilberto Axel Chang, PhD	361	Structural Engineering, Seismic Engineering	/citations?user=KGOMPHgAAAAJ&hl=es&oe=ASCII	KGOMPHgAAAAJ	361	213	2	2	2	1
7	Ildeman Abrego Castillo	329	Ciencias de los Materiales	/citations?user=74EDpbIAAAAJ&hl=es&oe=ASCII	74EDpbIAAAAJ	329	78	5	3	3	3
8	Héctor Montes Francovich	300	Robotica, Control	/citations?user=EpT2FMcAAAAJ&hl=es&oe=ASCII	EpT2FMcAAAAJ	300	168	11	8	12	6
9	Vladimir Villarreal	275	Ambient Intelligence, Ubiquitous Computing, Ambien...	/citations?user=52pqbIwAAAAJ&hl=es&oe=ASCII	52pqbIwAAAAJ	275	212	9	8	8	8
10	JOSE R. FABREGA-DUQUE	262	tropical hydrology, environmental chemistry, hydro...	/citations?user=R8E2jucAAAAJ&hl=es&oe=ASCII	R8E2jucAAAAJ	262	94	8	5	7	3
11	Norea L. Miller	207	Mapas conceptuales, pedagogías activas, aprendizaj...	/citations?user=uvyQt2AAAAAJ&hl=es&oe=ASCII	uvyQt2AAAAAJ	207	111	8	6	7	3
12	Cristian Iván Pinzón Trejos	197	Inteligencia Artificial	/citations?user=w7WcOwAAAAAJ&hl=es&oe=ASCII	w7WcOwAAAAAJ	197	182	7	6	6	5
13	Eleicer Ching Prado	180	Ciencia de materiales	/citations?user=BXdlDCAAAAAJ&hl=es&oe=ASCII	BXdlDCAAAAAJ	180	36	6	3	5	1
14	Humberto Rodriguez	167	control of robots, image processing, control of UAVs,...	/citations?user=d1FBqKIAAAAJ&hl=es&oe=ASCII	d1FBqKIAAAAJ	167	35	9	4	9	0
15	Adan Vega Saenz	157	soldaduras, elementos finitos, mecanica computacion...	/citations?user=O8fVFVIAAAAJ&hl=es&oe=ASCII	O8fVFVIAAAAJ	157	113	7	7	5	4
16	Ramiro Vargas	153	Structural Engineering, Seismic Design, Steel Structur...	/citations?user=BvEQzMIAAAAJ&hl=es&oe=ASCII	BvEQzMIAAAAJ	153	108	6	5	8	4
17	Congreso Clabes	144	Abandono, deserción, educación, congreso	/citations?user=vh-fDgAAAAJ&hl=es&oe=ASCII	vh-fDgAAAAJ	144	142	6	6	3	3
18	Pablo Montero	137	food technology, shelf life, food processing	/citations?user=y1vykoAAAAAJ&hl=es&oe=ASCII	y1vykoAAAAAJ	137	131	3	3	3	3
19	Casilda Saavedra	115	climate change, environmental sciences	/citations?user=U44S_kAAAAAJ&hl=es&oe=ASCII	U44S_kAAAAAJ	115	89	2	2	1	1
20	Carlos Vergara-Chen	89	Marine ecology, molecular ecology, aquatic ecologi...	/citations?user=23d0IQQAAAAJ&hl=es&oe=ASCII	r23d0IQQAAAAJ	89	85	5	5	3	3

Figura 6. Estructura de salida de los datos de perfiles en GS extraídos con el algoritmo

El método "2 algoritmo en R", es el mismo algoritmo, pero se incluyó la Universidad de la República del Uruguay (UDELAR) con 182 perfiles y 6388 publicaciones. El tiempo promedio de este método fue de 4 minutos, inferior al tiempo de los métodos evaluados anteriormente. En ambas pruebas con el algoritmo, el tiempo de Scraper de los perfiles es inferior al mejor tiempo de los métodos anteriores "Figura 7".

Tabla 4. Prueba de tiempo promedio de scraper de datos de gs utilizando diferentes métodos de web scraping.

		Tiempo Scraper (minutos)			
Método	**#Perfiles / #Publicaciones**	**Perfiles**	**Publicaciones**	**Total**	**Horas**
Local (Copiar/Pegar)	55 / 1400	35	466	501	8,21
Local Browser	55 / 1400	8	125	133	2,13
Local Software	55 / 1400	12	184	196	3,16
Online	55 / 1400	7	124	131	2,11
1 Algoritmo en R	55 / 1400	1	2	3	0,03
2 Algoritmo en R	76 / 2232	1	3	4	0,04

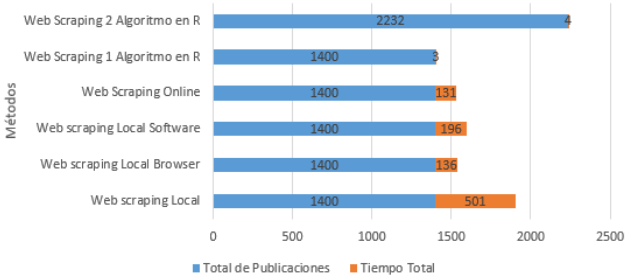

Figura 7. Gráfica de comparación de los Método de Scrapear de Perfiles y Publicaciones en GS.

4.2. Resultados de Scraper de los perfiles y publicaciones de 15 Universidades

Se realizó una prueba de extracción de todos los datos de 15 Universidades en GS utilizando el "Algoritmo en R": Universidad de la República (UDELAR), Universidad de Costa Rica (UCR), Université de Franche-Comté (UFC), Universidad de Antioquia (UDEA), Universidad de Chile (UCHILE), Universidad Nacional Autónoma de México (UNAM), Universidad de Osaka (OSAKAU), University of Edinburgh (UED), Universidad Politécnica de Valencia (UPV), University of Illinois at Urbana-Champaign (UILLINOIS).

Los resultados de las pruebas muestran que las Universidades con un promedio de 100 perfiles y 3400 publicaciones el tiempo de extracción fue de **2 minutos**. El tiempo total de scraper de 8364 perfiles y 175,086 publicaciones fue de **62 minutos**, inferior al tiempo de cualquier método aplicado , en las pruebas se extrajeron los perfiles, publicaciones y sus detalles "Tabla 5".

Tabla 5. Publicaciones por universidad en gs y tiempo de scraper con algoritmo en r.

Universidad	País	#Perfiles	#Publicaciones	Tiempo (minutos)
UFM	Guatemala	14	393	1
ESPOL	Ecuador	67	1061	1
FURB	Brasil	38	1360	1
UTP	Panamá	77	1434	1
UH	Cuba	79	2758	3
UDELAR	Uruguay	182	6388	2
UCR	Costa Rica	230	6952	2
UFC	Francia	119	7063	2
UDEA	Colombia	383	8429	3
UCHILE	Chile	566	11433	5
UNAM	México	1329	12670	11
OSAKAU	Japon	460	13038	4
UED	Escocia	1471	14091	14
UPV	España	794	29835	11
UILLINOIS	Estados Unidos	2555	58181	62
		8364	**175086**	**122**

4.3 Problemas en el uso del paquete "scholar"

El uso del paquete Scholar en el algoritmo de detalles permitió agilizar el desarrollo de este, sin embargo, encontramos que el paquete tenía un error al extraer detalles de publicaciones con más de 100 registros. En las primeras 9 Universidades mostradas en la "Tabla 5" donde se verificó de forma manual que los resultados de cantidad de perfiles y publicaciones es el indicado. En las otras 6 Universidades "Tabla 6" donde los perfiles tenían más de 100 publicaciones, estos perfiles se extrajeron con 0 publicaciones, al verificar los perfiles algunos casos tenían hasta 2000 publicaciones. En la extracción de datos de la UNAM de 566 perfiles solo se extrajeron 137 perfiles con el número de publicaciones correctas, en OSAKAU de los 1329 solo 140, de la UED 140 perfiles de 460, de la UPV 387 de 794 y de la ULLINOIS 2250 de 2555.

El problema que encontramos es que la función get_publications() del paquete "SCHOLAR" el cual extrae los detalles de las publicaciones, contiene una variable (FLUSH=false) que extrae los datos que están en el caché de GS, cuando se habilitó a (FLUSH=true), algunos perfiles que tenían valor de 100, cambiaron su valor a 1000 ó 2000 a la hora de volver hacer la extracción. Sin embargo, en algunos perfiles que había sido scrapeados de forma correcta, pasaron a tener 0 publicaciones, por lo que el valor de (FLUSH= true/false) no permite scrapear los datos de forma correcta cuyos perfiles tengan más de 100 publicaciones.

5. Conclusión

Según los resultados obtenidos, el Web Scripting resulta ser una alternativa funcional para extraer datos de un sitio web, sin embargo, con los métodos web online y de escritorio realizados a través de aplicaciones no logramos obtener el objetivo deseado, en tiempo y datos estructurados.

La opción de crear un algoritmo, aunque más compleja a la hora de desarrollarlo, es la mejor opción para obtener

datos personalizados, el tiempo de Scraper resulto inferior en las pruebas y el tiempo máximo de las 15 Universidades fue menor al de cualquier método utilizado. Con el algoritmo logramos extraer más datos de perfiles y publicaciones en menos tiempo, con detalles de las publicaciones y los datos que se almacenan en los ya están estructurados, por lo que permite un mejor análisis de estos.

La realización de este proyecto y la culminación de forma satisfactoria de esta etapa, será de gran beneficio para las Universidades involucradas en la medición de la producción científica y académica en la Red, ya que contarán con una herramienta para minimizar el trabajo de extracción de datos de GS y analizar el impacto de las publicaciones y perfiles.

6. Trabajos futuros

Se realizará cambios en el algoritmo utilizando programación vectorizada para minimizar el tiempo de ejecución del algoritmo, también creando una pausa entre la extracción de un perfil y otro para verificar si esto elimina el problema en la extracción de perfiles extensos.

Se creará un esquema para realizar un nuevo algoritmo para extraer los datos de los detalles de las publicaciones, eliminando el uso del paquete "Scholar" y personalizar los datos a extraer.

La visión de este proyecto no es solo realizar extracción de datos, sino hacer análisis, evaluación, visualización de los datos por lo que se incluirán funciones para ello. Se contempla extraer los perfiles y publicaciones de Universidades en GS de Centroamérica, Latinoamérica y los diferentes continentes para hacer análisis de los datos.

Enlace de Algoritmo Versión 1.0 utilizando el paquete "Scholar"
https://bitbucket.org/dannymu/ejemplos-de-r

7. Referencias

[1] A. M. VELÁZQUEZ, "Tim Berners-Lee: «El papel no desaparecerá, siempre habrá cosas que nos guste leer en ese formato»," 2012. [Online]. Available: http://www.lne.es/asturama/2012/02/15/tim-berners-lee-papel-desaparecera-habra-cosas-guste-leer-formato/1199452.html.

[2] M. Castells, "Internet y la Sociedad Red," *La Factoría*, vol. 14–15, pp. 1–12, 2001.

[3] M. F. Berners-Lee, "Weaving the Web. HarperOne," 1999.

[4] J. R. Sánchez Carballido, "Perspectivas de la información en Internet: ciberdemocracia, redes sociales y web semántica," *Zer-Revista Estud. Comun.*, vol. 13; n.° 25, pp. 61–81, 2011.

[5] L. C. Silva Ayçaguer, "El índice-H y Google Académico: una simbiosis cienciométrica inclusiva," *ACIMED*, vol. 23, no. 3, pp. 308–322.

[6] M. Oficial and E. N. Log, "Logística , Transporte Y Cadena De," 2014.

[7] D. Torres and Á. Cabezas, "Altmetrics : nuevos indicadores para la comunicación científica en la Web 2 . 0," pp. 53–60, 2013.

[8] D. I. Directions, T. Mining, U. K. Further, and H. Education, "The Value and Benefits of Text Mining," no. March, 2012.

[9] R. B. Penman and D. Martinez, "Web Scraping Made Simple with SiteScraper."

[10] F. Borrego, "Alternativas para realizar web scraping," 2017. [Online]. Available: http://felicianoborrego.com/alternativas-para-realizar-web-scraping/.

[11] R. Cotton, *Learning R, O'RELLY*. 2013.

[12] K. Cichini, "GScholarScraper_3.1," 2012. [Online]. Available: https://github.com/gimoya/theBioBucket-Archives/blob/master/R/Functions/GScholarScraper_3.1.R.

[13] J. Keirstead, "Package Scholar," 2015. [Online]. Available: https://cran.r-project.org/web/packages/scholar/index.html.

[14] Extension Google Chrome, "Scraper," 2015. [Online]. Available: https://chrome.google.com/webstore/detail/scraper/mbigbapnjcgaffohmbkdlecaccepngjd?utm_source=chrome-app-launcher-info-dialog.

[15] Fminer, "FMiner Scraping," 2015. [Online]. Available: http://www.fminer.com/.

[16] Import.io, "Import.io," 2016. [Online]. Available: https://www.import.io/.