Revista de +D TECNOLÓGICO
Portal de revistas: http://revistas.utp.ac.pa

Extracción de datos de perfiles en Google Scholar utilizando un algoritmo en el lenguaje R para hacer minería de datos

Web Scraping of profiles in Google Scholar using an algorithm in the R language to do data mining

Danny Murillo [1*], Dalys Saavedra [2], Erika Quintero [3]
[1,2] *Vicerrectoría de Investigación, Postgrado y Extensión, Universidad Tecnológica de Panamá, Panamá*
[3]*Facultad de Ingeniería de Sistemas Computacionales, Universidad Tecnológica de Panamá, Panamá*
***Autor de correspondencia:** danny.murillo@utp.ac.pa

RESUMEN– El objetivo de este artículo es hacer uso de la técnica Web Scraping para extraer datos de Google Scholar (GS)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. Además, con el fin de realizar un análisis más profundo, se creó 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. 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.

Palabras claves– *Web Scraping, Google Scholar, minería de datos, lenguaje R, análisis de datos.*

ABSTRACT– The purpose of this article is to show a study using the Web Scraping technique to extract data from Google Scholar through several methods. Web Scraping is a way of no strutured Data Miner which allow: to extract information from websites, to scan its HTML code and to generate patterns of data extraction. In addtion, to obtain better analysis in this study, an algorithm was created based on the R language in order to compare the speed of data extraction and the effciciency related to the format of out data as well as to identify a better way of extraction data from GS as structured way.

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

Citación: D. Murillo, D. Saavedra y E. Quintero, "Extracción de datos de perfiles en Google Scholar utilizando un algoritmo en el lenguaje R para hacer minería de datos", *Revista de I+D Tecnológico*, vol. 14, n.° 1, pp. 93-103, Jun. 2018.
Tipo de artículo: Original. **Recibido:** 17 de octubre de 2017. **Recibido con correcciones:** 27 de octubre de 2017. **Aceptado:** 11 de abril de 2018.

filosofía que Google, pero haciendo converger en una sola plataforma diferentes servicios [7].

Los datos de GS son relevantes para realizar análisis Bibliométrico. Es posible hace análisis de datos de: perfiles de investigadores, perfiles de revistas indexadas, número de citaciones que tienen las publicaciones en estos perfiles, como el impacto de estos perfiles a nivel mundial a través de su hindex. 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.

1.1 Minería de datos y Web Scraping

La necesidad de utilizar metodologías de análisis inteligente de datos que permitan descubrir patrones de datos para generar conocimiento útil, es la mejor forma de definir la minería de datos (MD). Esta se representa analógicamente como la extracción de pequeñas partículas de oro (información) en las minas de gran cantidad de rocas (datos), donde es necesario pulir el oro, para generar el verdadero valor (conocimiento) [8].

La búsqueda de este conocimiento en los datos utilizando la minería de datos está basado en el uso de la metodología KDD (Knowledge Discovery in Databases) o Descubrimiento del conocimiento a partir de datos. Este término que fue acuñado en 1989, se refiere a todo el proceso de extracción de conocimiento a partir de una base de datos, el cual marca un cambio de paradigma en el que lo importante es el conocimiento útil que seamos capaces de descubrir a partir de los datos [9] [10]. El proceso para descubrir el conocimiento KDD es un proceso interactivo y recursivo de siete pasos que se van dando a medida que se obtienen resultados preliminares que requieren replantear las variables iniciales. Los pasos de la metodología KDD son: Limpieza de datos, integración de datos, selección de datos, transformación y adaptación de datos, minería de datos, evaluación de patrones, generación de conocimiento mostrados en la figura 1 [11].

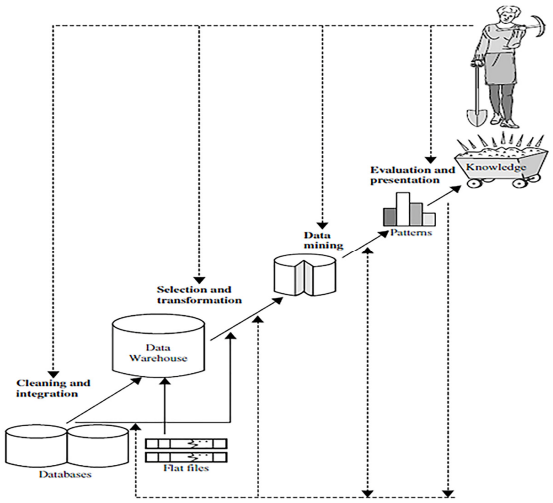


Figura 1. Paso del Proceso KDD (Knowledge Discovery in Databases).

La MD integra varias áreas como estadísticas, inteligencia artificial, base de datos y programación, y la minería de texto (MT). La MT identifica la información útil de los contenidos / datos / documentos de la web, sin embargo, tales datos en su forma más amplia deben ser reducidos a información útil. Los datos de contenido web consisten en datos estructurados como datos en las tablas, datos no estructurados como textos libres y datos semiestructurados como documentos HTML. Aquí, los diversos enfoques en la minería de contenido web están representados [12] [13].

La MT integra la minería de contenido web, que contiene 4 formas de extracción: minería de datos no estructurada, minería de datos estructurada, minería de datos semiestructurada, datos multimedia. En la "minería de datos estructurada" se encuentran las técnicas de "Web Crawler" que son rutinas o programas que analizan la estructura de una página web. "Page content mining" que funciona en las páginas clasificadas por los motores de búsqueda. En el caso de la técnica de Web Crawler incluye un técnica de extracción llamada "Web scraping" [14] [15]. Particularmente en este estudio se hará énfasis en la técnica Web Scraping. El "Web Scraping" es una técnica que consiste en la extracción del código HTML de una o varias páginas web de forma automatizada. El objetivo de extraer el código es evaluar su estructura y guardar los contenidos de forma estructurada, con datos que resulten con información relevante para su análisis posterior [16].

1.2 Análisis del proceso de Web Scraping

Scrapear los datos de un sitio web es una tarea común para buscadores como Google, Yahoo o Bing. Estos sitios realizan una tarea de "scaneo" a través de sus Spider o arañas buscando nuevas páginas web y documentos para su indexación. Este proceso analiza diariamente millones de páginas y documentos web con términos diferentes por página para ser indexados [17].

Esta tarea cotidiana para los motores de búsqueda puede ser restringida por algunas páginas web. De esta manera, su uso debe hacerse de forma responsable y metódica para que no se entienda que el proceso que se está realizando es malicioso, ya que al realizar el proceso de forma repetitiva a la misma página puede traer problemas como bloqueo al sitio web a través del IP. Por ello es necesario antes de hacer web scraping tomar en cuenta aspectos como[18]:

1. Comprobar que el archivo "robots.txt", no tiene el contenido restringido. *User-agent: * Disallow:*
2. Revisar las políticas de privacidad y acceso de la página web a scrapear.
3. No realizar tareas repetidas de scraping en corto tiempo, ya que puedo ser objeto de bloqueo.
4. Verificar que los contenidos o enlaces a extraer no hayan sido ocultos, a través de la capa de visualización o CSS:none; es una técnica llamada *honeypots* para detectar arañas web.

1.3 Objetivos del artículo

Este trabajo muestra el uso de varias herramientas de Scraping para extraer datos de los perfiles y publicaciones de GS. La idea de extraer estos datos surge de la necesidad de conocer el número de citaciones generadas por los artículos de las revistas de la UTP integrados en el Portal de Revistas e indexadas en Google Scholar en el 2016 [13]. El uso de estas herramientas, aunque permite extraer información estructurada del texto de un sitio web, no lo hacen realmente como está almacenado el contenido en la página. Esto se debe a que las operaciones que están implicadas en la extracción de información deben ser guiadas por un usuario o a través de un proceso autómata, algo que no es posible realizar con aplicaciones. Utilizando como base el lenguaje R queremos crear un algoritmo para reducir el tiempo de procesamiento y lograr una mejor estructuración de los datos al momento de la extracción de datos utilizando Web Scraping. R es un lenguaje de programación de código abierto, desarrollado por el grupo Core Team, además 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++[19].

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

2.1 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 [20].

2.2 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 [21].

2.3 Análisis de métodos de Web Scraping

Se realizó una evaluación de 4 métodos de web scraping para comparar y evaluar la velocidad de extracción y la personalización de la estructura de salida al extraer los datos de los perfiles y detalles de publicaciones en GS.

2.3.1 Selección de métodos

Realizamos las pruebas utilizando 4 métodos: copiar y pegar, *Local Browser*, *Local Software*, *Online*. En la tabla 1 se presenta un resumen de estos métodos. 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
Copiar y Pegar	Manual	ninguno	ninguno
Web Scraping Local Browser	Extensión Chrome	gratuito	técnico mínimo
Web Scraping Local Software	Fminer	pago	técnico intermedio
Web Scraping Online	Import.io	pago (Free versión)	técnico mínimo

2.3.2 Aplicaciones utilizadas para cada método

2.3.2.1 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.

2.3.2.2 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 [22].

2.3.2.3 Web scraping Local *Software*: FMiner es un *software* que permite abrir la página web en la aplicación y grabar el proceso de selección de nodos html, creando un diagrama de flujo de datos de la página web y asignando el valor seleccionado a cada variable, el proceso es semi-automático, ya que el usuario debe escoger cuales son los datos que desea guardar [23].

2.3.2.4 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 [24].

2.3.3 Datos utilizados en cada método

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). En la tabla 2 se muestra datos del número de perfiles y publicaciones que tenía cada Universidad en GS.

Tabla 2. Perfiles de universidades seleccionadas para web scraping en Goggle 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

2.3.4 Tiempo de Scraper de cada método

Para cada perfil de las universidades seleccionadas se aplicó cada método, donde se extrajeron todos los perfiles y las publicaciones de cada perfil, midiéndose 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** según tabla 3. En estas pruebas no se consideró extraer los detalles de cada publicación, por lo que el tiempo pudo ser mayor.

Tabla 3. 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, el objetivo de esta prueba era conocer el valor promedio de extracción de los datos de perfiles y publicaciones utilizando estos métodos. Conociendo que el mejor valor de extracción fue de **2 horas 18 minutos**, se realizó un análisis e implementación de un algoritmo en el lenguaje R para automatizar el proceso de extracción, disminuir el tiempo y generar datos estructurados.

3. Metodología

3.1 Recursos

Las aplicaciones, paquetes en R y características técnicas utilizadas en el análisis fueron:

- 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.
- 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.

3.2 Análisis de estructura de datos

Una de las características más utilizadas en la extracción de datos semiestructurada en páginas web es la "técnica basadas en árboles". La representación de una página Web mediante un etiquetado ordenando de sus nodos se conoce normalmente como DOM (Document Object Model). Cada nodo representa una etiqueta HTML y juntas puede representarse como ramas ordenadas y rotuladas. La jerarquía de árbol representa los diferentes niveles de anidamiento de los elementos que constituyen la página Web, la idea detrás del modelo es que las páginas web que contiene etiquetas HTML, se muestren como texto y palabras claves que puede ser interpretado por el navegador para representar links, botones, imágenes, tablas [25].

En la figura 2 se muestra la estructura del DOM de los bloques de los perfiles de la afiliación de la Universidad Tecnológica de Panamá (UTP) en GS, identificando patrones repetitivos en los datos como el nombre, citaciones, palabras claves.

Figura 2. Listado de perfiles de Google Scholar, afiliación Universidad Tecnológica de Panamá.

Para poder realizar la extracción fue necesario conocer el código HTML que compone cada bloque con datos del perfil en GS. En la figura. 3 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 3. 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 según tabla 4. Separamos cada uno de los nodos HTML que contenían los datos de los elementos individuales que serían almacenados en variables para luego agruparlas en una tabla en R llamada data.frame, esta permite almacenar diferentes tipos de datos.

Tabla 4. Scraper de datos por valor html y los resultados de cada bloque por variable

Variable	Valor html	Función en R	Resultado
url_perfil	https://scholar.google.es/citations?view_op=view_org&hl=es&org=4736061867397421563	read_html(url_GS)	Código html de la página web
afiliación	h2.gsc_authors_header	html_text(url_perfil, h2.gsc_authors_header)	Nombre de la Universidad
Perfil	div.gs_scl	html_node(url_perfil, div.gs_scl)	Código de Bloque de Perfil
nombre	h3>a	html_text(Perfil, h3>a)	Reinhard Pinzón
url_perfil	href	html_attr(Perfil,href)	https://scholar.google.es/citations?user=1TICxmUAAAAJ&hl=es
Id_user	https://scholar.google.es/citations?user=1TICxmUAAAAJ&hl=es	extraer_id(url_perfil)	1TICxmUAAAAJ&hl

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 parámetro de la clase identificada, R mostró los bloques de contenidos extraídos 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 su 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 Algoritmo para extraer los perfiles en GS

Se desarrolló un algoritmo para extraer todos los ID de usuarios de los perfiles de una afiliación utilizando la URL de afiliación de una Universidad en GS. En la figura 4 se muestra el esquema del algoritmo, donde se extrae 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, 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.

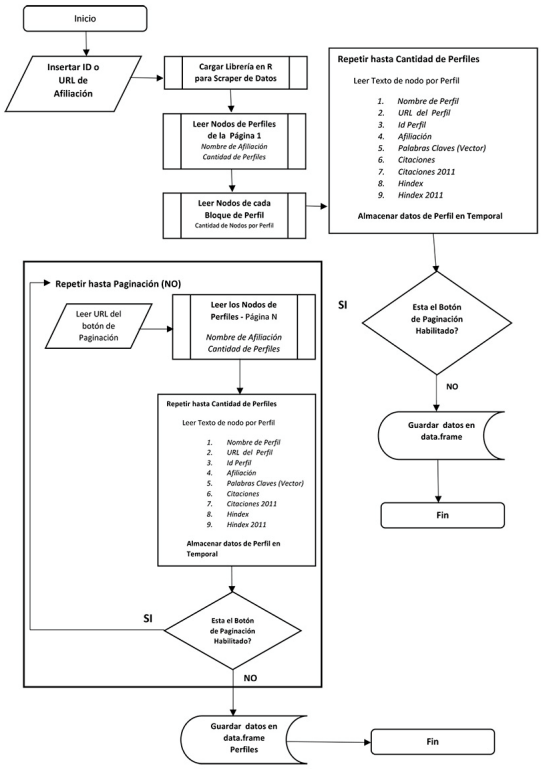


Figura 4. Esquema de Algoritmo para Scrapear datos de Perfiles de una Afiliación en Google Scholar.

3.5.2 Algoritmo para Scrapear detalles de Perfiles

En la figura 5 se muestra el esquema del algoritmo para extraer los detalles de los perfiles utiliza la URL de cada perfil que fue extraído en el algoritmo de extracción de perfiles. Se realiza una lectura de la cantidad de perfiles guardados y se lee el campo URL que contiene el enlace de cada perfil para empezar a leer los datos de cada perfil, los datos extraídos fueron: palabras claves del perfil, citaciones, citaciones 2011, hindex, hindex_2011, estos datos se vincularon a la URL del perfil y el ID del perfil. En este algoritmo se extrajo nuevamente las palabras claves, pero cada palabra fue almacenada separada por coma para su posterior análisis. Los datos fueron almacenados en data.frame temporal hasta finalizar el ciclo de repetición.

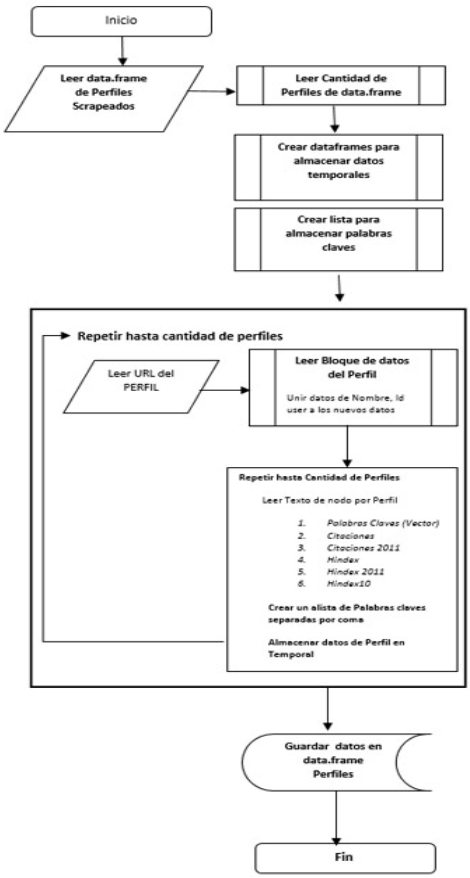


Figura 5. Esquema de algoritmo para Scrapear de detalles de los perfiles de una Afiliación en Google Scholar.

3.5.3 Algoritmo para extraer publicaciones en GS

Se creó un segundo algoritmo para extraer todas las publicaciones por perfil. En la figura 6 se muestra el uso del paquete S*cholar* de R y la función get_publications() que permitió extraer las publicaciones y sus detalles. 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. 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.

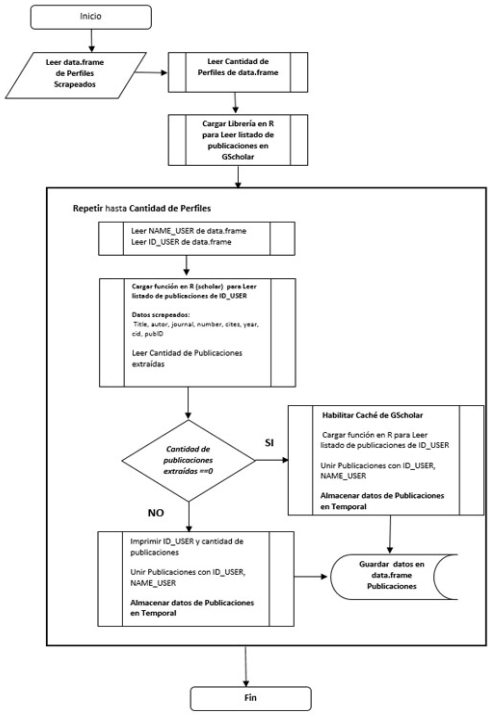


Figura 6. Esquema de Algoritmo para Scrapear todas las Publicaciones por perfil de una Afiliación en Google Scholar

4. Resultados

4.1 Comparación de métodos y algoritmo en R

Se realizó una evaluación del algoritmo en R utilizando los datos de las cinco universidades anteriores con 55 perfiles y 1400 publicaciones. Las pruebas utilizando el algoritmo en R, método "1 algoritmo en R" es de tres minutos incluyendo perfiles, publicaciones y detalles de las publicaciones la cual se muestra en la tabla 4. El algoritmo generó los datos extraídos de forma estructurada en R, que fueron exportados al formato .CSV y MS Excel, el esquema se muestra en la figura 7.

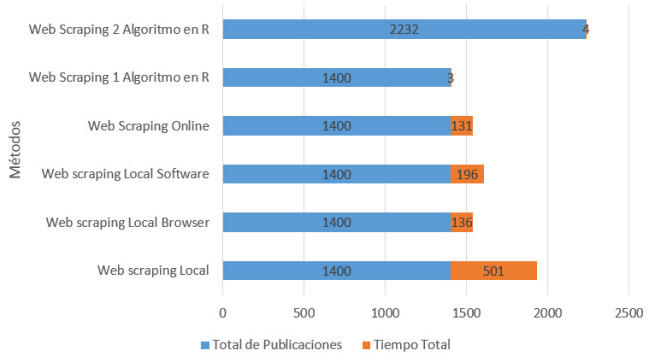
	Nombre	word_key	url_user	id_user	citaciones	cita_2011	hindex	hindex_2011
1	Reinhardt Pinzon Adames		/citations?user=1TICxmUAAAAJ&hl=es&oe=ASCII	1TICxmUAAAAJ	935	452	10	6
2	Elida de Obaldia	Material Science, Renewable Energy, Diamond thin fil...	/citations?user=I8gpxI4AAAAJ&hl=es&oe=ASCII	I8gpxI4AAAAJ	880	272	9	5
3	Maritza Morales Batista	inteligencia artificial, open data, linked data, big data,...	/citations?user=UdcSsuYAAAAJ&hl=es&oe=ASCII	UdcSsuYAAAAJ	467	219	11	8
4	Oscar M. Ramirez	Structural and Earthquake Engineering	/citations?user=Ko4jMoMAAAAJ&hl=es&oe=ASCII	Ko4jMoMAAAAJ	426	259	6	6
5	Rodney Delgado-Serrano	Astrophysics, Astronomy, Physics, Mathematics, Info...	/citations?user=8IUF9cMAAAAJ&hl=es&oe=ASCII	8IUF9cMAAAAJ	413	234	6	6
6	Gilberto Axel Chang, PhD	Structural Engineering, Seismic Engineering	/citations?user=KCGMPHgAAAAJ&hl=es&oe=ASCII	KCGMPHgAAAAJ	381	213	2	2
7	Ildenan Abrego Castillo	Ciencias de los Materiales	/citations?user=74B2jt0AAAAJ&hl=es&oe=ASCII	74B2jt0AAAAJ	329	76	5	3
8	Héctor Montes Franceschi	Robotica, Control	/citations?user=EpT2FMcAAAAJ&hl=es&oe=ASCII	EpT2FMcAAAAJ	300	168	11	8
9	Vladimir Villarreal	Ambient Intelligence, Ubiquitous Computing, Ambien...	/citations?user=52pqkbwAAAAJ&hl=es&oe=ASCII	52pqkbwAAAAJ	275	212	9	8
10	JOSE R. FABREGA-DUQUE	tropical hydrology, environmental chemistry, hydro...	/citations?user=RiE2jucAAAAJ&hl=es&oe=ASCII	RiE2jucAAAAJ	262	94	8	5
11	Norma L. Miller	Mapas conceptuales, pedagogías activas, aprendizaj...	/citations?user=uxvQt2AAAAAJ&hl=es&oe=ASCII	uxvQt2AAAAAJ	207	111	8	6
12	Cristian Iván Pinzón Trejos	Inteligencia Artificial	/citations?user=w7Wc8w4AAAAJ&hl=es&oe=ASCII	w7Wc8w4AAAAJ	197	102	7	6
13	Eleicer Ching Prado	Ciencia de materiales	/citations?user=EKdDC4AAAAJ&hl=es&oe=ASCII	EKdDC4AAAAJ	180	36	6	3
14	Humberto Rodriguez	control of robots, image processing, control of UAVs,...	/citations?user=dIF8qX8AAAAJ&hl=es&oe=ASCII	dIF8qX8AAAAJ	167	35	9	4
15	Adan Vega Saenz	soldadura, elementos finitos, mecanica computacion...	/citations?user=O8fkfvYAAAAJ&hl=es&oe=ASCII	O8fkfvYAAAAJ	157	113	7	7
16	Ramiro Vargas	Structural Engineering, Seismic Design, Steel Structur...	/citations?user=Br4QoIMAAAAJ&hl=es&oe=ASCII	Br4QoIMAAAAJ	153	108	6	5
17	Congreso Clabes	Abandono, deserción, educación, congreso	/citations?user=vfr-FbgAAAAJ&hl=es&oe=ASCII	vfr-FbgAAAAJ	144	143	6	6
18	Pablo Montero	food technology, shelf life, food processing	/citations?user=y1vykoAAAAAJ&hl=es&oe=ASCII	y1vykoAAAAAJ	137	131	3	3
19	Casilda Saavedra	climate change, environmental sciences	/citations?user=UI4EB_8AAAAJ&hl=es&oe=ASCII	UI4EB_8AAAAJ	115	89	2	2
20	Carlos Vergara-Chen	Marine ecology, molecular ecology, aquatic ecologi...	/citations?user=z3s60QAAAAJ&hl=es&oe=ASCII	z3s60QAAAAJ	89	85	5	5

Figura 7. 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 cuatro minutos, inferior al tiempo de los métodos evaluados anteriormente. En la gráfica 1 se muestra ambas pruebas con el algoritmo, el tiempo de Scraper de los perfiles es inferior al mejor tiempo de los métodos anteriores.

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



Gráfica 1. Comparación tiempo de los métodos de Scrapear de perfiles y publicaciones en GS.

4.2 Resultados de Scraper de las 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).

En la tabla 5 se muestran los resultados de las pruebas de las universidades con un promedio de 100 perfiles y 3400 publicaciones el tiempo de extracción fue de dos **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. 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	Japón	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 seis universidades según 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 habían sido scrapeados de forma correcta, pasaron a tener 0 publicaciones, por lo que el valor de (FLUSH= true/false) no permite extraer los datos de forma correcta cuyos perfiles tengan más de 100 publicaciones.

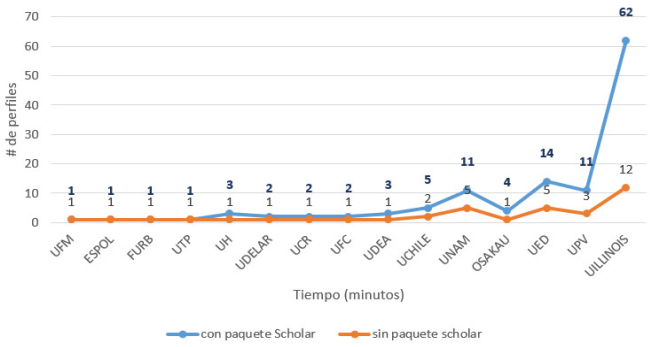
4.4 Resultados entre Algoritmos utilizando paquete Scholar y sin usar el paquete

Para probar si realmente el uso del paquete Scholar en el algoritmo de extracción de perfiles era el problema de los malos datos de extracción, se realizó una función en R llamada "PubGS_publications_get" para extraer los datos de los perfiles antes de realizar el recorrido de todos los perfiles de afiliación.

En la tabla 5 se muestra el total de perfiles que fueron contabilizados de forma manual por cada Universidad. En la tabla 6 se muestra los perfiles y el tiempo del algoritmo utilizando el paquete Scholar y el algoritmo con la nueva función reemplazando el paquete, los resultados muestran que el nuevo algoritmo no solo extrajo el total de perfiles contabilizados por universidad, sino que el tiempo de extracción por cada universidad fue menor principalmente en las 6 universidades cuyos perfiles no fueron extraídos de forma correcta. El tiempo promedio de extracción por perfil fue de **1.66 segundos** (algoritmo con paquete Scholar), **0.37 segundos** (nuevo algoritmo sin paquete Scholar), el tiempo total de extracción de los perfiles fue de 38 minutos (nuevo algoritmo sin paquete Scholar) y 122 minutos (algoritmo con paquete Scholar) sin el total de perfiles. En la gráfica 2 se muestra la comparación de tiempo entre los dos algoritmos y cada universidad.

Tabla 6. Perfiles extraídos por universidad en Gs utilizando paquete "Scholar" y algoritmo sin paquete Scholar

	Con paquete Scholar		**Sin paquete Scholar**	
Universidad	**#Perfiles**	**Tiempo (minutos)**	**#Perfiles**	**Tiempo (minutos)**
UFM	14	1	14	1
ESPOL	67	1	67	1
FURB	38	1	38	1
UTP	77	1	77	1
UH	79	3	79	1
UDELAR	182	2	182	1
UCR	230	2	230	1
UFC	119	2	119	1
UDEA	383	3	383	1
UCHILE	**137**	5	**566**	2
UNAM	**138**	11	**1329**	5
OSAKAU	**140**	4	**460**	1
UED	**147**	14	**1471**	5
UPV	**387**	11	**794**	3
UILLINOIS	**2250**	62	**2555**	12
	4388	**122**	**8364**	**38**



Gráfica 2. Comparación de tiempo en minutos de extracción de los perfiles en GS utilizando paquete Scholar y algoritmo sin el paquete Scholar.

4.5 Problemas al scrapear datos de GS

Aunque se siguieron las indicaciones de la metodología para scrapear datos en el sitio web, en varias ocasiones GS bloqueo nuestra IP al extraer datos.

En algunas ocasiones por lapsos de 1 hora y en otras por un periodo de 1 día. GS nos mostró que fuimos bloqueados de la siguiente forma: campo de CAPTCHA en el sitio web, Error 403 Forbidden, Error 503 Service Unavailable, 401 Unauthorized, 404 Not Found, 408 Request Timeout, 429 Too Many Requests.

Para evitar el bloqueo se creó un listado de tiempos aleatorio entre 15 segundos y 60 segundos para que fueran asignados al extraer cada perfil. También se modificó en función “read_html” que lee el código html a scrapear, modificando el parámetro “useragent” de forma aleatoria, como si fueran diferentes usuarios que acceden con diferentes navegadores, los agent utilizados fueron: "Mozilla/5.0, Trident/6.0)", "Opera/9.80.

Otras opciones que para mejorar el scraper, aunque no fueron aplicadas son: crear un servidor Proxy, rotar el número de IP, crear un algoritmo autónomo capaz de hacer un scaneo de datos de forma estructurada pero aleatoria cada vez que realice el proceso. Otro proceso que si se aplicó fue almacenar los datos scrapeados en un documento, luego cargar el documento y estructurar los datos, esto evita realizar constante consultas al servidor de Google.

5. Conclusiones

Con los resultados obtenidos, el Web Scraping resulta ser una alternativa funcional para extraer datos de un sitio web, lo cual no se logra obtener con los métodos web online y de escritorio utilizados comúnmente.

La opción de crear un algoritmo, aunque más compleja a la hora de desarrollarlo, fue 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 se logró extraer más datos de perfiles y publicaciones en menos tiempo y los datos fueron estructurados mientras se extraían por lo que permite realizar un mejor análisis de estos.

Las pruebas realizadas entre la versión del algoritmo utilizando el paquete scholar y el nuevo algoritmo, permite mostrar que siempre existe mejoras que se pueden realizar no solo en la optimización del proceso de Web Scraping, sino en la reducción del tiempo de extracción. Estas nuevas pruebas permitieron identificar algunos elementos a considerar en esta técnica para evitar ser bloqueados por los sitios web donde se van a extraer los datos, tomando en cuenta siempre las políticas de extracción de la página.

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 realizarán 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 algoritmo autómata para poder modificar el proceso de escaneo y mejorar el proceso aleatorio de asignación de tiempo para evitar posibles bloqueos de GS.

El proyecto no solo busca hacer 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 más detallado y comparativo por país y región.

Enlace del código del Algoritmo en el Lenguaje R: 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] UIAF, "Técnicas de minería de datos para la detección y prevención del lavado de activos y la fi nanciación del terrorismo (LA/FT)," p. 35, 2014.
[9] J. C. Riquelme, R. Ruiz, and K. Gilbert, "Minería de datos: Conceptos y tendencias," *Intel. Artif.*, vol. 10, no. 29, pp. 11–18, 2006.
[10] R. B. Penman and D. Martinez, "Web Scraping Made Simple with SiteScraper."
[11] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.
[12] D. I. Directions, T. Mining, U. K. Further, and H. Education, "The Value and Benefits of Text Mining," no. March, 2012.
[13] D. S. Danny Murillo, "Implementación de Plataforma Digital de Revistas Académicas y Científicas electrónicas en la Universidad Tecnológica de Panamá para mejorar su visibilidad a nivel nacional e internacional," in *Tecnología, innovación e investigación en los procesos de enseñanza-aprendizaje*, 2016, pp. 936–947.
[14] S. Shi, C. Liu, Y. Shen, C. Yuan, and Y. Huang, "AutoRM: An effective approach for automatic Web data record mining," *Knowledge-Based Syst.*, vol. 89, pp. 314–331, 2015.
[15] V. Bharanipriya and V. K. Prasad, "Web Content Mining Tools : a Comparative Study," *Int. J. Inf. Technol. Knowl. Manag.*, vol. 4, no. 1, pp. 211–215, 2011.
[16] F. Borrego, "Alternativas para realizar web scraping," 2017. [Online]. Available: http://felicianoborrego.com/alternativas-para-realizar-web-scraping/.
[17] M. Peshave, "How Search Engines Work and a Web Crawler Application," 2010.
[18] Scrapehero, "Scalable do-it-yourself scraping – How to build and run scrapers on a large scale," 2015. [Online]. Available: https://www.scrapehero.com/scalable-do-it-yourself-scraping-how-to-build-and-run-scrapers-on-a-large-scale/.
[19] R. Cotton, *Learning R, O´RELLY*. 2013.
[20] K. Cichini, "GScholarScraper_3.1," 2012. [Online]. Available: https://github.com/gimoya/theBioBucket-Archives/blob/master/R/Functions/GScholarScraper_3.1.R.
[21] J. Keirstead, "Package Scholar," 2015. [Online]. Available: https://cran.r-project.org/web/packages/scholar/index.html.
[22] Extension Google Chrome, "Scraper," 2015. [Online]. Available: https://chrome.google.com/webstore/detail/scraper/mbigbapnjcgaffohmbkdlecaccepngjd?utm_source=chrome-app-launcher-info-dialog.
[23] Fminer, "FMiner Scraping," 2015. [Online]. Available: http://www.fminer.com/.
[24] Import.io, "Import.io," 2016. [Online]. Available: https://www.import.io/.
[25] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowledge-Based Syst.*, vol. 70, pp. 301–323, 2014.