Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza

ISSN: 2525-1333
Vol.:2-Nro.1 (Agosto-2017)

Informe técnico

DESARROLLO DE SOFTWARE PARA LA CAPTURA DE REVISTAS CIENTÍFICAS EDITADAS A TRAVÉS DE OJS (OPEN JOURNAL SYSTEMS)

DEVELOPMENT OF SOFTWARE FOR THE CAPTURE OF SCIENTIFIC JOURNALS EDITED THROUGH OJS (OPEN JOURNAL SYSTEMS)

Juan José Marenco [(1)]

[(1)]Docente Investigador Universidad Nacional de La Matanza
jjmarenco@gmail.com

ISSN: 2525-1333. Vol.:2-Nro.1 (Agosto-2017)

Resumen:

Este desarrollo tecnológico tiene como eje central contar con una herramienta de cosecha de los contenidos de las revistas científicas realizadas con el software OJS. En esta primera etapa, el desarrollo capturará los artículos de las revistas pertenecientes a Nuestra Universidad que se editan por medio del citado soft. Esta captura incluirá los datos principales para poder contar con una base de datos de todos los artículos desarrollados por estas revistas. El software final apunta a contar, en un segundo artículo, con un sistema que realice la misma función pero con todas las revistas científicas que se encuentren en Internet, logrando así, una base de datos que podría usarse como un metabuscador de directorios científicos pero exclusivamente de publicaciones abiertas.

Abstract:

This technological development is a tool for harvesting the contents of scientific journals made with OJS software. In this first stage, the development will capture the articles from Our University contained in magazines made with OJS soft. This capture will include the main data in order to have a database of all the articles developed by these magazines. The final development –planned for a second article- will try to harvest the magazines present in some system of Information on scientific research journals. So the user of the software will have a database that could be used as a search engine for scientific directories but exclusively for open publications.

Palabras Clave: *OJS, base de datos, artículos*

Key Words: *OJS, database, articles*

ISSN: 2525-1333. Vol.:2-Nro.1 (Agosto-2017)

I. CONTEXTO

La recolección de datos sobre los contenidos de las revistas hechas en base al software Open Journal Systems (de ahora en más OJS) pueden ser almacenados con propósitos de abastecer necesidades propias de cualquier Institución como así también, para contar con la posibilidad de ser un agente de visualización de los aportes científicos en revistas de contenido abierto. En este último caso no es difícil pensar en transformar un sitio en metabuscador de los datos en dichas publicaciones

II. INTRODUCCIÓN

Aunque el nacimiento de las revistas científicas datan de fines del Siglo VXII, el principio de su creación no es tan diferente al que hoy las genera. Estas podrán cambiar los soportes de comunicación, pero la premisa sigue siendo ser una forma de comunicación rápida y de una amplia divulgación. Para que esto pueda cumplirse, obviamente, toda revista debe contar con pautas programáticas y de edición que jerarquicen cada publicación[1]. Varias son las características que tanto editor comoautor del artículo deben cumplir para alcanzar esta meta de calidad. Dentro de estas pautas, sin lugar a dudas, la visibilidad del artículo es importante. Lograrlo dependerá, primero, de los niveles de indización de la revista y, segundo, de la visibilidad y accesibilidad que posea por sí misma la propia revista [2]. Este artículo está dirigido a aumentar esa segunda opción ya que el soft que se presenta lograr capturar los datos de localización y el texto completo del artículo para luego ser usado tanto por el propietario de los contenidos como por terceros interesados en la difusión de los textos.

III. MÉTODOS

Se utilizó Python®para la realización de la aplicación en su versión 2.6 ya que es se trata un medio de codificación muy sencillo y de fácil adaptación al tratamiento de textos y generación de bases de datos. El software (cuyas líneas de códigos se disponen completas en el titulo siguiente) plantea capturar de las revistas generadas con OJS en la UNLaM los siguientes datos:

1- Autor/es de los artículos.
2- Título de los mismos.
3- Dirección http:// donde se encuentra dispuesto.
4- El texto plano completo del artículo originalmente en formato PDF.

Para lograr esto solo debe contarse con la dirección de inicio de la revista analizada ya que toda las líneas analizan las paginas en su formato de código fuente para llegar a los datos antes citados. Aunque en la UNLaM se cuenta hasta el presente con tres revistas científicas, solo dos están realizadas con OJS. Ellas son la Revista de Investigación del Departamento de Humanidades y Ciencias Sociales (Rihumso) y la Revista Digital del Departamento de Ingeniería (ReDDI).

Para una mejor visualización de las líneas de código y comprender los pasos en que se desarrolló el software, se presenta el siguiente diagrama (Fig.: 1).

ISSN: 2525-1333. Vol.:2-Nro.1 (Agosto-2017)

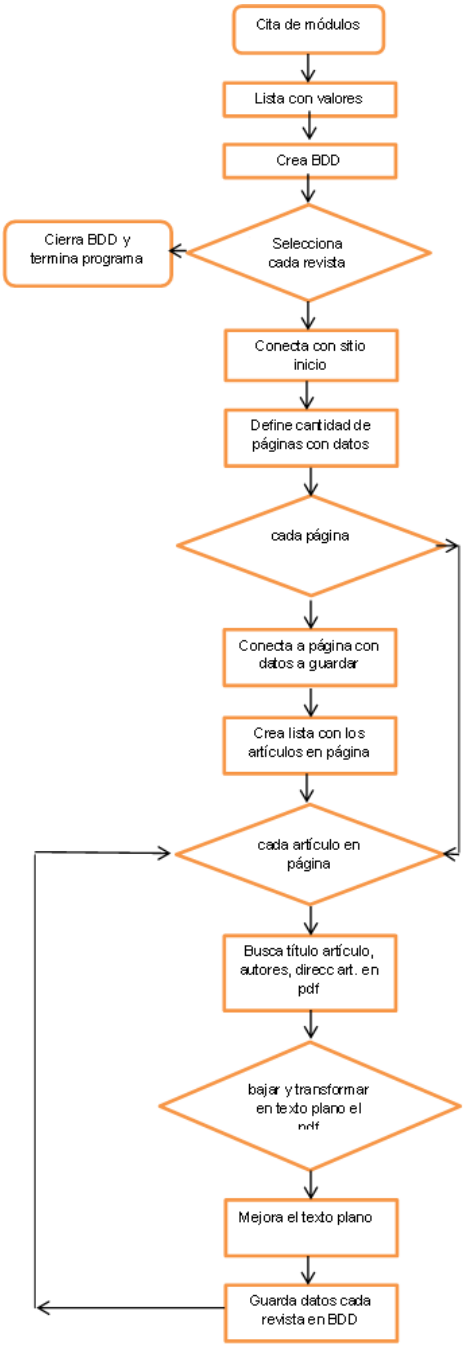

Fig. 1: diagrama del programa

IV. RESULTADOS Y OBJETIVOS

Se presenta a continuación las líneas de código del programa en Python® que es totalmente funcional. Debe hacerse notar que en el caso de trabajarse en SO Windows en la carpeta donde se tenga este programa se tiene que contar con el módulo pdf2txt.pyc y una carpeta con la descarga del módulo pdfminer como se puede ver la captura de pantalla. (Fig.: 2)

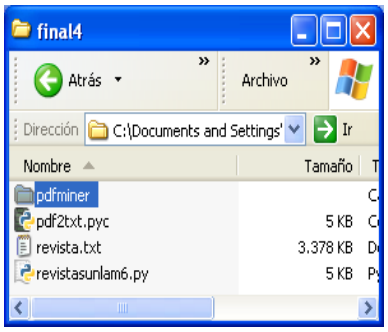

Fig. 2: distribución de los archivos en la carpeta

En el archivo "revista.txt" -que es generado por el programa- se guardará la base de datos obtenida, en el archivo llamado "revistasunlam6.py" se encuentra la codificación que se propone, y los demás elementos son los comentados en el párrafo anterior y que son necesarios para transformar los PDF's en texto plano.

En tabla 1 se puede obtener el código fuente completo y funcional.

ISSN: 2525-1333. Vol.:2-Nro.1 (Agosto-2017)

TABLA 1
Codificación utilizada

```
#!C:\Python26\python.exe
#-*- coding: utf-8 -*-
############
fromcStringIOimportStringIO
frompdfminer.pdfinterpimportPDFResourceManager, PDFPageInterpreter
frompdfminer.converterimportTextConverter
frompdfminer.layoutimportLAParams
frompdfminer.pdfpageimportPDFPage
import os, sys, urllib2, warnings, pyPdf, urllib
warnings.simplefilter('ignore', DeprecationWarning)
valores=['rihumso','humanidades','RIHS','reddi','reddi','REDI']
# el 1º y 4º elemento de lista son lo que inicia la direc web (primerenlink)
# el 2º y 5º la carpeta en la web de búsqueda (segundoenlink)+
# el 3º y 6º son los codigos iniciales para guardar en la base de datos y
# reconocer si el articulo es de una de las dos
baserevista=open('revista.txt','w')
def revista(primerenlink, segundoenlink, codigorev):
    #pagina 1
nrorevista=1
canthojas =
urllib2.urlopen('http://'+primerenlink+'.unlam.edu.ar/index.php/'+segundoenlink+'/search/search?authors=
***&searchPage=1#results')
leecantidadhojas= canthojas.read().replace('\n','').replace('\t','')
canthojas.close()
    # cuantas paginas tiene
ifleecantidadhojas.count('"#results">&gt;&gt;</a>"')==0:
cantidadpaginas=1
listapaginas=[1]
else:
cantidadpaginas=int(leecantidadhojas.split('"#results">&gt;&gt;</a>"')[0].split('searchPage=')[-1])
listapaginas=[]
forvalue in range(1,cantidadpaginas+1):
listapaginas.append(value)
fornropag in listapaginas:# repite la extracción hasta llegar al final de las páginas totales
printnropag
        htmlredi1 =
urllib2.urlopen('http://'+primerenlink+'.unlam.edu.ar/index.php/'+segundoenlink+'/search/search?authors=
***&searchPage='+str(nropag)+'#results')
        leeredi1= htmlredi1.read().replace('\n','').replace('\t','')
htmlredi1.close()
        # se pone una bandera para separacion de los artículos
        cadaart1=leeredi1.replace('"<tr><tdcolspan="3"
class="separator"> </td></tr><trvalign="top"><td><a
href="','puntodeencuentroderevistas').replace('"<tr><tdcolspan="3"
class="headseparator"> </td></tr><trvalign="top"><td><a href="','puntodeencuentroderevistas')
        cadaart2=cadaart1.split('puntodeencuentroderevistas')
for i in range(1,len(cadaart2)):
```

ISSN: 2525-1333. Vol.:2-Nro.1 (Agosto-2017)

TABLA 1 (continuación)

```
        # se analiza cada bandera  para extraer los elementos dirección web  (direcweb)
        # tiitulo del libro (titulomagazine), autor o autores (autoresmagazine)
        # y con direccionbajada se busca primero descargar el PDF
        # y transformarlo en texto plano (interior3)
direcweb=cadaart2[i].split('"" class="file">PDF"')[0].split('"href="')[-1].replace('view','download')
ifdirecweb.count('HTML')<>0:
pass
else:

titulomagazine=cadaart2[i].split('"<tdwidth="30%">"')[1].split('"</td>"')[0]
autoresmagazine=cadaart2[i].split('"<tdcolspan="3" style="padding-left: 30px;font-style:
italic;">"')[1].split('"</td>"')[0]
            ######### texto completo del pdf pero plano
            #descarga y guarda pdf
direccionbajada=urllib.urlretrieve (direcweb, "bajada.pdf")
            # lo lee y lo transforma en texto plano
pages=[]
pagenums = set(pages)
output = StringIO()
manager = PDFResourceManager()
converter = TextConverter(manager, output, laparams=LAParams())
interpreter = PDFPageInterpreter(manager, converter)
infile = file('bajada.pdf', 'rb')
try:
for page in PDFPage.get_pages(infile, pagenums):
interpreter.process_page(page)
infile.close()
converter.close()
text = output.getvalue()
except:
text='no se ha podido extraer el texto- revise el pdf desde el sitio'
output.close
testo = text.replace('\n','').replace('\t','').replace('&',' ')
            ##########
            interior1=testo.replace('$',' ')
            interior2=interior1.replace('!',' ')
            interiora2=interior2.replace('#',' ')
            interior3=interiora2.replace(chr(39),'')
            baserevista.writelines('('+chr(39)+codigorev+str(nrorevista)+chr(39)+',
'+chr(39)+autoresmagazine+chr(39)+', '+chr(39)+titulomagazine+chr(39)+',
'+chr(39)+direcweb+chr(39)+', '+chr(39)+interior3+'),'+'\n')
nrorevista+=1
printcodigorev+str(nrorevista),direcweb,'***','hoja',nropag
for val in range(0,5,3):
revista(valores[val],valores[val+1],valores[val+2])
baserevista.close()
```

ISSN: 2525-1333. Vol.:2-Nro.1 (Agosto-2017)

V. DISCUSIÓN

El producto ha sido probado y es totalmente funcional logrando una base con los siguientes campos en cada registro:

1º código definido por RIHS+número secuencial y por REDI+número secuencial en el caso de la revista de Humanidades y de Ingeniería respectivamente.
2º Autor/res
3º Título
4º dirección del artículo en pdf
5º todo el contenido del pdf en formato de texto plano sin retornos de carro ni tabulaciones

Cada vez que se ejecute el archivo "revistasunlam6.py" y ha habido algún cambio en una dirección de un artículo editado tiempo atrás, o se hayan agregado artículos nuevos, estos nuevos registros reemplazarán los datos almacenados con anterioridad teniendo una base totalmente actualizada.

VI. CONCLUSIONES

El sistema propuesto en estas páginas puede ser usado para poder contar con una base de datos operativa para agilizar las búsquedas de textoincluido en los artículos. Este contenido permite que otrasinstituciones cuenten con una base de datos de las revistas científicas de la UNLaM. Al mismo tiempo, las líneas de código pueden, con algunos cambios, ser útiles para la generación de bases a partir de otras revistas.

VII. REFERENCIAS Y BIBLIOGRAFÍA

[1] M.Patalano. "Las publicaciones del campo científico: las revistas académicas de américa latina", ANALES DE DOCUMENTACION, Nº 8, PÁGS. 217-235,2005

[2] D. Fernández Quijada. "Revistas científicas e índices de impacto. A propósito de 'Hacer saber' ", ÁREA ABIERTA Nº 20, Referencia: AA19. 0807.104, JULIO 2008

Recibido: 2017-08-18
Aprobado: 2017-08-23
Datos de edición: Vol. 2 - Nro. 1 - Art. 4
Fecha de edición: 2017-08-23
URL: http://www.reddi.unlam.edu.ar