UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

PCVN

Presentado por Roberto Poza Puras
en Universidad de Burgos — 11 de febrero de 2019
Tutor: Dr.César Ignacio García Osorio
y Dr. Juan Jose Rodriguez Diez

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D.César Ignacio García Osorio y D. Juan Jose Rodriguez Diez, profesores del departamento de Ingeniería Civil, Área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Roberto Poza Puras, con DNI 71291761H, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado PCVN.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 11 de febrero de 2019

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. César Ignacio García Osorio	D. Juan Jose Rodriguez Diez

Resumen

La labor de investigación de los profesores es una tarea difícil y que requiere de mucho esfuerzo y que no siempre recibe el valor que le corresponde. Aún después de este trabajo falta el proceso de acreditación por parte de las instituciones competentes, proceso el cual requiere de un nuevo esfuerzo por reunir todas estas investigaciones desde los distintos sitios donde fueron publicadas para someterlas al proceso pertinente.

PCVN lo que pretende es automatizar y simplificar de cara al usuario este proceso, ofreciendo una herramienta sencilla, pero a su vez también eficaz. Obteniendo la información de las principales bases de datos de investigaciones y subiendo toda esta información obtenida directamente al organismo competente. En este caso la ANECA en su aplicación ACADEMIA.

Descriptores

Web-Scrapping, Investigacion , Scopus, Web of Science, Google Scholar , ANECA , ACADEMIA ...

Abstract

The research work of teachers is a difficult task that requires a lot of effort and not always receive the value it deserves. After all this work, there is one more step, the accreditation process made by the competent institutions, a process that requires a new effort to group all the investigations made from the different sites where they were published to submit them to the accreditation process. PCVN aims to automate and simplify this process for the user, offering a simple tool, but at the same time effective. Obtaining information from the main research databases and uploading all this information directly to the competent institutions. In this case ANECA in its application ACADEMIA

Keywords

Web-Scrapping, Investigation , Scopus, Web of Science, Google Scholar , ANECA , ACADEMIA ...

Índice general

Índice de figuras

Índice de tablas

Introducción

El trabajo de investigación y desarrollo es una de las principales labores de los miembros de la universidad, un trabajo que permite la adquisición de nuevos conocimientos y teorías sobre el entorno que nos rodea mediante la interacción directa del sujeto con este. Estos conocimientos son muy valiosos pues no pueden ser adquiridos de ninguna otra forma, es por esto por lo que se debe apoyar la investigación en los ámbitos universitarios.

Este trabajo requiere de tiempo y esfuerzo para ser llevado a cabo y debidamente redactado y plasmado. Para después ser publicado en las revistas de divulgación científica, universidades etc. Tratando así de dar a conocer los nuevos hallazgos o conclusiones extraídas de la investigación. Es frecuente que los investigadores deseen acceder a los cuerpos docentes universitarios, pues se entiende que desean compartir los conocimientos adquiridos a través del tiempo y la investigación, para ello deben pasar por un proceso de evaluación y acreditación por la *Agencia Nacional de Evaluación y Acreditación* (*ANECA*). Este proceso implica aún más tiempo y esfuerzo, en la labor de reunir y tratar toda la información referente a las publicaciones realizadas por un autor, para posteriormente subir toda esa información al programa que para ello tiene habilitado la *ANECA* (*ACADEMIA*).

ACADEMIA lleva a cabo el proceso de evaluación curricular para la obtención de la acreditación para el acceso a los cuerpos docentes universitarios de Profesor Titular de Universidad y Catedrático de Universidad. Incluye el procedimiento para la exención del requisito de pertenecer al Cuerpo de Profesores Titulares de Universidad a que se refiere el art. 60.1 de la Ley Orgánica 6/2001, de 21 de diciembre[3]. El procedimiento de acreditación tiene abierta la presentación de solicitudes a través de la Sede Electrónica del MECD[2], mediante el uso de una aplicación informática [1], la cual permite

esencialmente la presentación de solicitudes y la cumplimentación del *CV*. En esto último es en lo que se enfoca principalmente este proyecto, en ayudar a los investigadores a reunir adecuadamente los datos bibliográficos referentes a las publicaciones realizadas para posteriormente realizar una subida de estos datos a la propia aplicación de la *ANECA* , con el correspondiente formato. Evitando así este tedioso proceso al usuario.

Así en este proyecto podemos diferenciar dos funciones

- Recopilar: Para recopilar la máxima cantidad de información posible sobre las publicaciones de un autor, consultaremos las principales bases de datos del ámbito de la investigación.
  - Google Scholar[9]
  - Scopus[6]
  - Web of Science[10]
- Enviar: Esta función no se compondrá únicamente de enviar la información a la aplicación de la *ANECA*, si no también de agrupar toda la información obtenida , dotarla de una estructura y formato adecuado y finalmente enviarla a esta aplicación.

Objetivos del proyecto

A continuación se pasará a detallar los objetivos que han motivado la realización de este proyecto, tanto a nivel global como técnico y personal.

2.1. Objetivos globales

- Automatizar el proceso de extracción de los datos bibliográficos de las bases de publicaciones anteriormente mencionadas.
- Dotar a la información bibliográfica extraída del formato propio de estos documentos (*BibTeX*,*RIS*,*EndNote*).
- Automatizar el proceso de subida de los datos extraídos a la aplicación *ACADEMIA*.
- Desarrollar una aplicación de escritorio que permita al usuario una fácil interacción con los procesos automatizados.
- Almacenar la información extraída para un posible uso futuro.

2.2. Objetivos técnicos

- Desarrollar una aplicación utilizando *Python* para la extracción de datos, así como la subida de estos a la aplicación.
- Utilizar *Scrum* como metodología de planificación del proyecto.
- Utilizar *Git* como sistema de control de versiones junto con la plataforma *GitHub*.

- Utilizar LaTeX como herramienta de documentación.
- Realizar test unitarios.

2.3. Objetivos personales

- Realizar una aportación al sector de la investigación en la universidad.
- Adquirir conocimientos en nuevas materias no tratadas durante la carrera.
- Explorar nuevos conceptos de trabajo y desarrollo.

Conceptos teóricos

Para la total compresión del proyecto es necesario tener claros algunos conceptos como son Técnica *Web Scraping*, formato *BibTex*, la herramienta *Selenium* y el protocolo *HTTP*.

3.1. Web Scraping

El Web Scraping es una técnica de extracción de información de una página web utilizando para este propósito programas software, su traducción inmediata al castellano sería algo así como “raspado web".[36] Esta técnica simula la navegación de un ser humano en la red, se puede hacer de varias formas:

- Utilizar el protocolo *HTTP*3.4 manualmente.
- Utilizar un navegador incrustado en el propio software, simulando la navegación tradicional que haría cualquier usuario tradicional.

Usando cualquiera de las dos alternativas arriba planteadas el objetivo es siempre el mismo, transformar los datos contenidos en una página web (normalmente con formato *HTTML*) en datos relevantes para ser almacenados y tratados con un fin.

En este proyecto lo que vamos a realizar es extraer la información relevante acerca de las publicaciones de un determinado autor de las principales páginas web del sector (Google Scholar, Scopus y Web of Science)

3.2. BibTex

BibTex es una herramienta software que permite dar formato a un texto, tradicionalmente listas de referencias y que es usado habitualmente junto con los documentos en LaTeX[24].

Es también un formato de archivo basado en texto, usado para definir datos bibliográficos (articulo,libros, ponencias en congreso etc.) habitualmente terminan en .bib y se caracteriza por que los elementos bibliográficos están separados por tipos. A continuación, se van a exponer los tipos más relevantes:

- *@article* artículo publicado en una revista. Campos Necesarios: author, title, journal, year, volume Campos Opcionales: number, pages, month, doi, note, key
- *@book* libro publicado con un editor concreto. Campos Necesarios: author/editor, title, publisher, year Campos Opcionales: volume/number, series, address, edition, month, note, key, url
- *@inproceedings* Articulo presentado en una conferencia o congreso. Campos Necesarios: author, title, booktitle, publisher, year Campos Opcionales: editor, volume/number, series, type, chapter, pages, address, edition, month, note, key
- @inbook Parte de un libro, suele ser un capítulo cseccion, etc. Campos Necesarios: author/editor, title, chapter/pages, publisher, year Campos Opcionales: volume/number, series, type, address, edition, month, note, key

@article{ ISI:000454418300026,
Author = {Tang, Yufei and Liu, Zhaowei and Zhao, Kang},
Title = {Fabrication of hollow and porous polystyrene fibrous membranes by electrospinning combined with freeze-drying for oil removal from water},
Journal = {JOURNAL OF APPLIED POLYMER SCIENCE},
Year = {2019},
Volume = {136},
Number = {13},
Month = {APR 5},
Publisher = {WILEY},
ISSN = {0021-8995},

Times-Cited = {0},
}
Imagen de ejemplo de una publicación en formato BibTeX.
Para saber más sobre los distintos tipos y campos que admite este formato[25].

3.3. Selenium

Selenium WebdriverEs una herramienta software de código abierto, que proporciona un entorno de pruebas para aplicaciones web, permitiendo realizar las pruebas en cualquier navegador.

A pesar de que tiene un entorno de desarrollo integrado (IDE), también posee librerías para su uso en los lenguajes de programación mas usados (*Java, C#,Ruby,Groovy,Perl,Php y Python*).Además es multiplataforma lo que permite que pueda ser utilizado en los distintos sistemas operativos, a través de la mayor parte de navegadores (*Google Chrome, Internet explorer,Firefox,Safari,Opera,HtmlUnit,phanmjs,Android,IOS*)[20]

A pesar de que Selenium dispone de varios componentes, el componente que nos interesa es Selenium WebDriver

- *Selenium web driver* a diferencia de su antecesor *Selenium RC* no necesita de un servidor especial para ejecutar las pruebas, si no que iniciará una instancia del navegador elegido y lo controlará, permitiendo al usuario navegar de una forma similar a como lo haría cualquier usuario convencional.

3.4. Protocolo HTTP

Protocolo HTTP o protocolo de transferencia de hipertexto[29] es el protocolo de comunicación que rige las comunicaciones en la red.

El protocolo HTTP se basa en un modelo de petición y respuesta, en la que el usuario realiza una petición y el servidor responde a la petición realizada, normalmente estas peticiones van acompañadas de parámetros o argumentos necesarios para que el servidor procese la petición y genere una respuesta.

Existen distintos tipos de métodos para interactuar, pero los fundamentales para la comprensión del funcionamiento de este proyecto son *GET y POST*.

- El método *GET* realiza una petición sobre un recurso específico, devolviendo información, en ningún caso debería tener otro efecto. Ejemplo de petición *GET*:
  GET / HTTP/1.1
  Host: ubuvirtual.ubu.es
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8*
  Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
  Accept-Encoding: gzip, deflate
  Referer: https://www.google.com/
  DNT: 1
  Connection: close
  Cookie: MoodleSessionmoodlecurrent=pnscbpi2tuv0anl6s4lciu8181
  Upgrade-Insecure-Requests: 1

- El método *POST* envía una serie de datos para que sean procesados por el recurso al cual se le está haciendo la petición, como consecuencia puede resultar en la modificación de los recursos del servidor. Los datos deberán ser incluidos en el cuerpo de la petición.

  Ejemplo de petición *POST*

  POST /login/index.php HTTP/1.1
  Host: ubuvirtual.ubu.es
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8*
  Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
  Accept-Encoding: gzip, deflate
  Referer: https://ubuvirtual.ubu.es/
  Content-Type: application/x-www-form-urlencoded
  Content-Length: 33
  DNT: 1
  Connection: close
  Cookie: MoodleSessionmoodlecurrent=8rv7dpjrg9n34dnqmds0aprgb3
  Upgrade-Insecure-Requests: 1

  username=example&password=example

Se define como un protocolo sin estado, esto quiere decir que no guarda ninguna información de otras conexiones. Es por esto que surge el concepto de *cookies*[26], que es una pequeña cantidad de información que se almacena en el equipo del usuario y que permite recordar si el cliente ya ha accedido al servidor anteriormente, permitiendo mostrar una y otra información o creando el concepto de *sesión* [34]. Así cuando iniciamos sesión en una página web con nuestro usuario y contraseña lo que se genera es una *cookie*, y cada vez que se envíe una petición al servidor esta es enviada para indicar que estamos autenticados con eso usuario y contraseña.

Técnicas y herramientas

En este apartado se van a exponer todas aquellas técnicas y o herramientas que han sido utilizadas durante la realización del proyecto. Se expondrá una pequeña definición junto con una explicación para que han sido utilizados y por qué se eligió esta herramienta y no otra.

4.1. Lenguaje de Programación

Python

Como lenguaje de programación se ha utilizado Python en la versión más reciente en la que se encontraba al inicio del proyecto 3.7.1.Actualmente se encuentran en la versión 3.7.2.

Es uno de los lenguajes más extendido, por su facilidad para aprender a programar, por su simplicidad, su extensa comunidad lo que hace más fácil aún aprender y solucionar los problemas encontrados y por las librearías que han sido creadas para este lenguaje lo que hace muy fácil trabajar con ciertos tipos de datos, importarlos /exportarlos etc. Además, cabe destacar que es de código abierto y multiplataforma[32].

Se eligió este lenguaje para el desarrollo del proyecto, por las librerías que posee, en especial para realizar las labores de *web scraping* y el tratamiento de los datos bibliográficos.

4.2. Entornos de Desarrollo

Jupyter NoteBook

Juypter es una aplicación web de código abierto que permite la creación y ejecución de código abierto. Se puede utilizar mediante el navegador sin necesidad de instalar absolutamente nada o bien instalando con *Anaconda* o *pip*4.4.

Es aplicación es ideal para el marco de pruebas de concepto, pues no necesita un desarrollo muy extenso de la idea, si no que al ser ejecución en vivo permite desarrollar el concepto de una forma ágil e intuitiva[13].

PyCharm

PyCharm es un entorno de desarrollo (IDE) específico para *Python* desarrollado por la empresa checa JetBrains. El cuál posee un gran número de herramientas y opciones para mejorar el proceso de desarrollo del código (escritura, revisión, comentarios , etc.)[31]

Se eligió este IDE pues es el más completo que existe para el lenguaje elegido y por qué disponía de licencia de estudiante, la cual permite acceder a todas las funcionalidades del entorno sin tener que pagar nada.

Al principio se planteó la idea utilizar *Sublime Text 3* como entorno de desarrollo, pero tras conocer el programa de estudiante para Pycharm, se pasó a utilizar esta herramienta.

Sublime Text

Sublime Text 3 es un editor de código capaz de interpretar un gran número de lenguajes adaptando la interfaz dependiendo del lenguaje, es una herramienta muy versátil, pero no tan completa como un IDE específico, a pesar de que dispone de numerosas herramientas y plugin para ayudar en el desarrollo del proyecto.[21]

Al principio se utilizó esta herramienta como entorno para el desarrollo del proyecto, pero se cambió a *Pycharm* por las razones ya comentadas anteriormente. Tras Esta decisión, se utilizó sublime para la realización de los ficheros *Markdown* que definen los *Sprints*.

4.3. Control de Versiones

GitHub

GitHub es una plataforma web para el hospedaje de repositorios, ha sido siempre la más usada y conocida. A lo largo de la carrera se ha trabajado en varias ocasiones con ella. Esto unido al hecho de que dispongan de licencia para estudiantes fue clave para decidir la herramienta a utilizar. Cabe destacar que es gratuita para proyectos de código abierto.[28]

Se barajó también la posibilidad de usar *Bitbucket*, pero por comodidad y por la licencia se utilizó GitHub.

4.4. Administrador de paquetes

pip

pip es un sistema de administración de paquetes software para *Python*.En *Pyhton 2.7.9* y posteriores ,así como en *Python 3.4* y posteriores se encuentra incluido por defecto.[30]
Utiliza una interfaz a través de la linea de comandos , lo que le hace muy sencillo de usar. Las opciones que permite son las siguientes :

- install
  pip install nombre-paquete
- uninstall
  pip uninstall nombre-paquete
- instalar lista de requerimientos
  pip install -r requisitos.txt
- instalar paquete para una versión de *Python* específica reemplazando *${versión}* por la versión correspondiente
  pip${versión} install nombre-paquete

4.5. Documentación

Texmaker

Texmaker es un editor gratuito para LaTeX que contiene la mayoría de las herramientas necesarias para la edición y desarrollo de un documento

LaTex, cabe destacar que posee auto corrector y auto-completado.[35]

4.6. Bibliotecas

Aquí vamos a mostrar las distintas Bibliotecas que se han usado a lo largo del proyecto y su función. Cabe destacar que todas ellas son para *Python*.

Scholarly

Scholarly[14] es un módulo que permite recuperar información referente a autores y publicaciones de *Google Scholar*, de una manera sencilla y amigable. Puede ser fácilmente instalada a través de *pip*4.4

Python-Scopus

Python-Scopus es una librería que interactúa directamente con la *API* de *Scopus*, haciendo esta interacción más sencilla y amigable para el usuario. Finalmente, se tuvo que desestimar el uso de esta librería pues por algún cambio no devolvía en su totalidad los autores correspondientes a un autor[38].

Bibtexparser

Bibtexparser es una librería dedicada a la carga y tratado de los ficheros con formato BibTeX.Incluye métodos para leer y escribir en fichero los datos almacenados en una lista de diccionarios, en la que cada diccionario es una publicación en la que la *"clave"* es el campo de la publicación (autor, titulo, etc.) y el *"valor"* será el valor de dicho campo[17].

Selenium Webdriver

Selenium Webdriver es una librería que permite crear una instancia del navegador elegido y controlarlo mediante código, consiguiendo así automatizar el proceso de navegación simulando que fuera un usuario corriente[20].

re

re es una librería para *Python* que permite el uso de las expresiones regulares, permitiendo una mejor extracción de la información relevante

de grandes cadenas de texto. Es una herramienta muy habitual en *"web scraping"*[8].

Tkinter

Tkinter es librería repleta de funcionalidades para el desarrollo de una interfaz gráfica en Python. Está orientada a objeto y aunque no es la única es la más utilizada, por su sencillez de uso y rapidez para dotar a una aplicación de una interfaz gráfica[22].

Unittest

Unittest test es un *framework* para Python que permite el uso y creación de pruebas unitarias[23].

requests

requests[16] es una librería para *HTTP*escrita en *Python* que permite una integración totalmente transparente de los servicios web, permitiendo realizar todo tipo de peticiones *HTTP* y manejando las respuestas recibidas por parte del servidor. Además, permite la reutilización de *keep-alive* y conexión *HTTP* automática gracias a urlib3

Pickle

Pickle es un librería para *Python* que permite serializar mediante protocolos binarios la estructura de un objeto *Python* , el proceso es reciproco es decir , se puede convertir un objeto en un flujo de bytes y se puede reconstruir el objeto a partir de un flujo de bytes[15].

4.7. Otras Herramientas

Burp Suite

Burp Suite es una herramienta gráfica para probar la seguridad de las aplicaciones web, pero en este proyecto ha sido utiliza para realizar ingeniería inversa a la aplicación web de la *ANECA(ACADEMIA)*. Permitiendo estudiar cómo se realizaban las peticiones *POST* y *GET* al servidor que información enviaban etc. Para ello se utilizó la herramienta *HTTP PROXY*, esta redirige todo el tráfico del navegador a la propia aplicación analizando todo el tráfico y permitiendo analizar que peticiones y respuestas recibe.

Photoshop

Photoshop es un editor de gráficos dedicado principalmente para el retoque de fotografías y gráficos, pero en este caso ha sido utilizado para el diseño del logo de este proyecto.

JabRef

JabRef es una herramienta software de gestión bibliográfica, la cual utiliza de formato nativo. Ha sido utilizada para la visualización y edición de los ficheros *BibTeX* generados a partir del *Web Scraping*.

Forticlient

Forticlient es una herramienta software que permite la conexión *SSLVPN* entre el dispositivo y una red,creando una conexión completamente encriptada y será enviada a través de un túnel seguro. También ofrece herramientas para mantener seguro nuestro equipo.

Aspectos relevantes del desarrollo del proyecto

En este apartado vamos a recoger los aspectos importantes que han ocurrido a lo largo del desarrollo del proyecto. Incluyendo las decisiones tomadas, los cambios en el proyecto o los problemas que hayan podido surgir y las soluciones que se han planteado (si se consiguió solucionar.)

5.1. Inicios del proyecto

La idea de trabajar en este proyecto de la búsqueda de ampliación de los conocimientos adquiridos a lo largo de la carrera. Ya poseía algún conocimiento superficial sobre el Web Scraping y sus métodos, es por eso por lo que parecía una idea interesante profundizar en esta materia. Además, la idea propuesta por el tutor se postulaba como adecuada a mis necesidades y asequible en los marcos de tiempo que se disponía.

Tras formalizarse los objetivos del proyecto y las metodologías de trabajo se pone en marcha el proyecto el *5/11/18*.

Figura 5.1: Logo de PCVN.

5.2. Metodologías

En la formalización del proyecto se utilizaría una metodología ágil basada en Sprints(Metodología Scrum)[19], de desarrollo incremental con revisiones.

- Se estableció que la duración de los Sprints sería de una semana
- Los objetivos para el inicio se marcarían semanalmente, revisando lo conseguido en el sprint anterior
- Se realizarían reuniones semanales, (siempre y cuando las circunstancias lo permitieran).

En líneas generales creo que los resultados de esta metodología han sido satisfactorios y que han tenido un impacto muy positivo en el proyecto.

5.3. Toma de decisiones

Páginas Web

En la formalización del proyecto también se declaró cual serían las páginas de las cuales se procedería a extraer la información. Por su relevancia y número de datos almacenados se decidió que se usaría:

- Google Scholar.
- Scopus.
- Web of Science.

5.4. Formato de almacenamiento

Una de las decisiones más importantes a tomar en el proyecto, es como se guardaría toda la información recolectada para ser posteriormente tratada. Se plantearon tres alternativas (BibTeX,EndNote,RIS)

BibTeX .

En la imagen (5.2) podemos ver un ejemplo de la información bibliográfica de una publicación mas concretamente un libro con los campos más comunes

```
@Book{abramowitz+stegun,
  author =  "Milton Abramowitz and Irene A. Stegun",
  title =   "Handbook of Mathematical Functions with Formulas,
        Graphs, and Mathematical Tables",
  publisher =   "Dover",
  year =    1964,
  address =     "New York",
  edition =     "ninth Dover printing, tenth GPO printing",
  isbn =    "0-486-61272-4"
}
```

Figura 5.2: Imagen de ejemplo de una publicación en formato BibTeX.Imagen extraída de [24].

EndNote.

En la imagen (5.3) vemos un ejemplo de la información bibliográfica referente a un artículo científico publicado en una revista en formato *EndNote*.

```
%0 Journal Article
%A Herbert H. Clark
%D 1982
%T Hearers and Speech Acts
%B Language
%V 58
%P 332-373
```

Figura 5.3: Imagen de ejemplo de una publicación en formato EndNote.Imagen extraída de [27].

RIS.

En la imagen (5.2) vemos la información bibliográfica de nuevo un artículo científico publicado por *Bell System Technical Journal* en formato *RIS*.

```
TY  - JOUR
AU  - Shannon, Claude E.
PY  - 1948/07//
TI  - A Mathematical Theory of Communication
T2  - Bell System Technical Journal
SP  - 379
EP  - 423
VL  - 27
ER  -
```

Figura 5.4: Imagen de ejemplo de una publicación en formato RIS.Imagen extraída de [33].

Todos son formatos de bibliografía basados en etiquetas y valor, en las que la etiqueta representa un campo (Ej. Autor, Título) y el valor de dicho campo (Ej. Roberto Poza, PCVN)

Aunque todos los formatos podrían ser perfectamente almacenados en la estructura de diccionarios de *Python*, finalmente se optó por usar el formato BibTeX pues las etiquetas correspondientes al campo son más explicativas y se adaptaban mejor a la propia estructura devuelta por las librerías, además las librerías para trabajar con este formato eran más numerosas y con amplia experiencia y documentación.

5.5. Librerías para la extracción de datos

Tras una toma de contacto con las técnicas de *Web Scraping* y las páginas de donde se sacaría esta información se procedió a determinar de qué forma se podría trabajar con esta información.

- Scholarly: Para la obtención de los datos de *Google Scholar*[14].
- Python-Scopus: Para la obtención de los datos de *Scopus*[38].
- Selenium: Para la obtenciones los datos De *Web of Science*[20].

5.6. Interfaz de usuario del proyecto

Se tenía la idea de dotar al proyecto de una interfaz, para que el contacto con el usuario no tuviera que ser a través de la línea de comandos. Se barajó la posibilidad de utilizar una interfaz basada en texto *TUI* o bien una interfaz gráfica *GUI*, finalmente se optó por esta última pues el esfuerzo extra que requería para su realización era aceptable y se entiendo que merecía la pena para ofrecer una mejor experiencia al usuario. Finalmente se usó la librería Tkinter para desarrollar esta interfaz, que si bien es sencilla es bastante funcional.

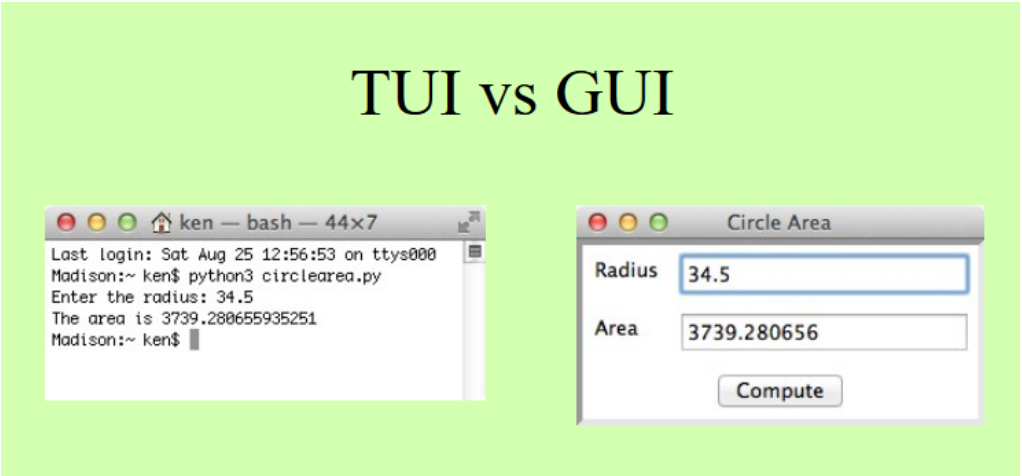

Figura 5.5: Imagen de ejemplo de una interfaz TUI vs GUI.Imagen extraída de [37]

5.7. Problemas encontrados

Error en función de Scholarly

En un principio se pretendió implementar una función que permitiera calcular cuantas de las citas de un artículo no eran auto citas, para esto se debía localizar cual eran las publicaciones que citaban la actual publicación y reconocer si alguno de los autores de la publicación citada se encontraba en la publicación citadora. Sin embargo la función encargada de devolver los datos de las publicaciones citadoras parecía sufrir algún tipo de problema pues no siempre funcionaba correctamente, como resultado se acabó descartando la idea dejando el número de citas tal y como lo devolvía *Google Scholar*

Diferencias entre formatos y duplicados

El archivo *BibTeX* con los datos referentes a *Web of Science* es generado directamente por la propia página web con lo que carecía del mismo formato que el generado por la librería *Bibtexparser*, es por eso que se tuvo que crear una función para el tratado de los datos, teniendo que renombrar algunos campos , asi como aplicar filtros de caracteres a otros. Además se aprovechó para a su vez aplicar la detección de publicaciones duplicados por doble factor: Titulo e ISSN.

Índice de calidad

Las publicaciones científicas se pueden dividir de acuerdo a si están indexadas de acuerdo con un índice de calidad relativo o no. En el caso de esta ultimas no existe problema, con los datos extraídos valdría sin embargo para las primeras es necesario más información (*Índice de impacto, posición de la revista en la categoría, tercíl, cuartil*).En la figura (5.6) se muestra las tablas de las que se extraerá la información ya mencionada. Para llevar a cabo es extracción se tuvo que utilizar la herramienta Selenium para buscar esta información en *JCR InCites*[11] para cada una de las revistas de la lista de publicaciones, se implementó un sistema de listas y diccionarios en el que una vez consultada los datos de una revista se almacenaba la información de todos los años, por si hiciera falta para otra publicación consiguiendo así reducir ligeramente los tiempos de ejecución pero aun así eran bastante elevados.

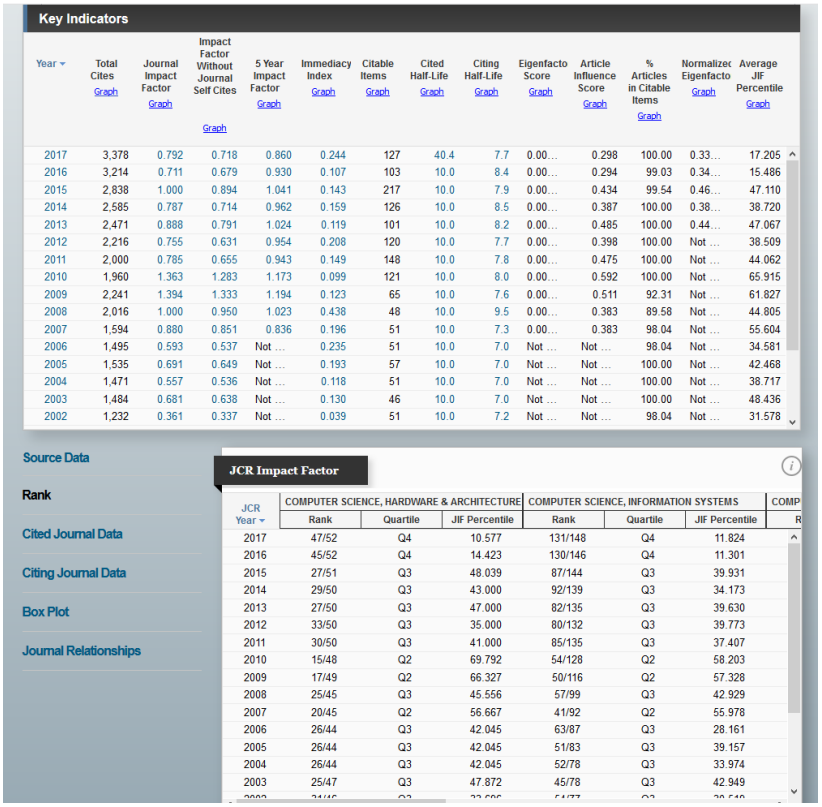

Figura 5.6: Imagen de ejemplo de las tablas contenedoras de los indicios de calidad de una publicación.Imagen extraída de [12]

Tiempo de ejecución en la consulta de índices de impacto

Como se ha comentado en el apartado anterior, los tiempos de ejecución eran aún muy elevados a pesar del sistema de listas y diccionarios implementados. El problema radicaba en que esa información recogida una vez finalizada era perdida, la solución fue bastante simple: guardar el objeto Python para que en la siguiente ejecución pudiera ser cargada y utilizada usando la librería *pickle*4.6. Con esta sencilla solución se consiguió reducir notablemente los tiempos de ejecución.

Error en Python-Scopus

Por algún motivo los resultados arrojados por la librería no estaban completos, en la mayoría de los casos no devolvía los autores de cada

publicación, se investigó las peticiones que realizaba y recibía la propia librería, pero no parecía haber nada erróneo por lo que se dedujo que debía ser algún problema con la gestión de las peticiones del propio servidor. Haciendo necesaria una nueva forma de obtener la información. Tras una nueva investigación sobre Scopus, se descubrió una opción muy parecida a la que presenta *Web of Science* para exportar la información encontrada a un fichero *BibTeX*, así que se decidió usar de nuevo la herramienta *Selenium* para buscar y exportar la información referente al autor.

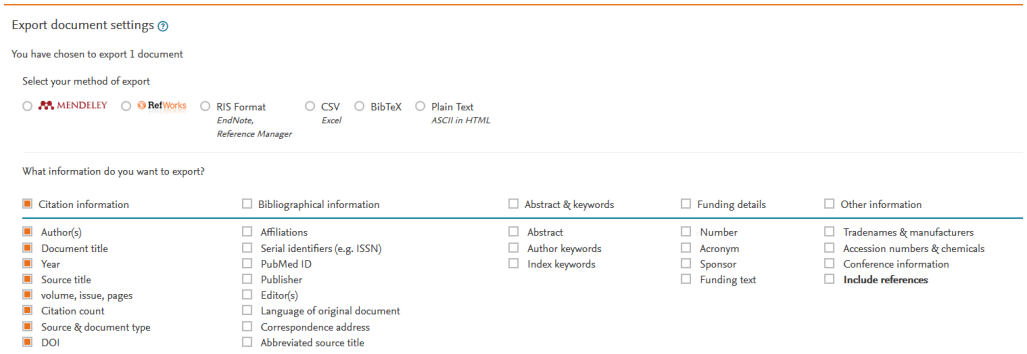

Figura 5.7: Imagen que muestra la opción para exportar los datos encontrados al formato elegido en Scopus.Imagen extraída de [6]

Errores con Selenium

Al realizar el proceso de subida de los datos extraídos y tratados a *ACADEMIA* se producían números fallos de carga de página y otra excepciones , lo que hacía que fallase la ejecución sin posibilidad de recuperarse y finalizando la ejecución de la aplicación. Además, el tiempo que tomaba era demasiado alto. Es por eso que se decidió investigar sobre como el navegador se relacionaba con el servidor para tratar de imitarlo directamente en la aplicación pasando a manejar directamente a mano las peticiones *POST* y *GET* sin necesidad de usar Selenium.

Para ello se utilizo la herramienta *Burp*4.7 , de la cual ya hemos hablado antes y que permite actuar de proxy entre el navegador y el propio servidor, viendo así que peticiones se realizan , con que parámetros y a que direcciones como se puede apreciar en la imagen (**??**). Se comenzó así un proceso de Ingeniería Inversa para replicar las interacciones generadas por la navegación normal

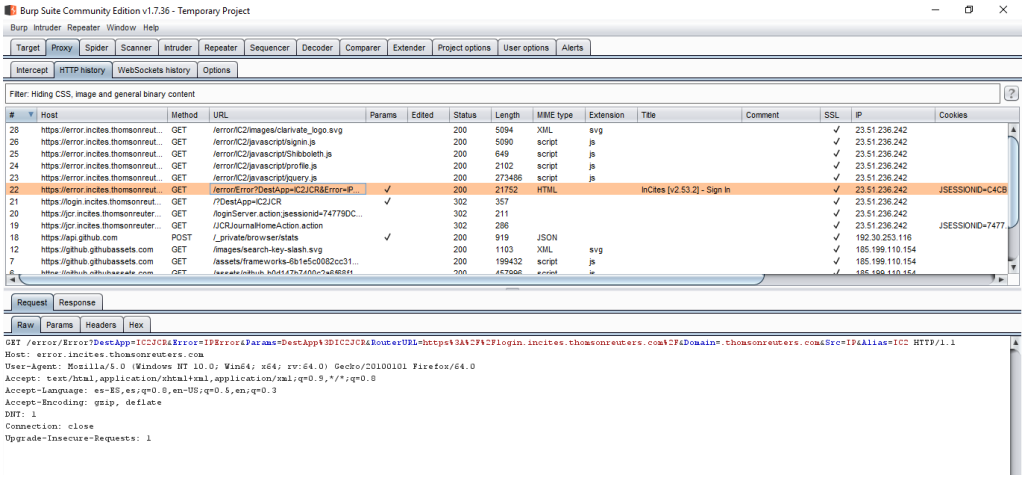

Figura 5.8: Ejemplo del análisis de capturas realizado con *Burp*.

El primer paso fue estudiar cuales eran los parámetros para el inicio de sesión en la aplicación para realizar la petición *POST* junto con el usuario y la contraseña correspondiente, tras esto se da un proceso de redirección hasta llegar a la pantalla principal de la aplicación. Como podemos ver en la imagen (5.9)

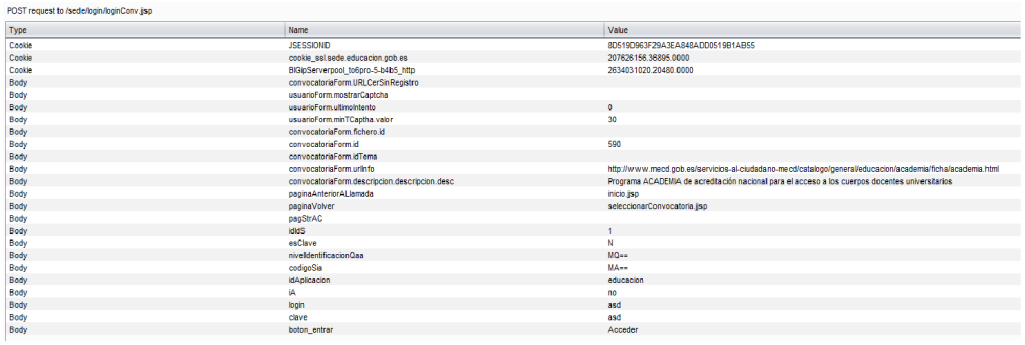

Figura 5.9: Análisis de la peticion *POST* para iniciar sesión.

El segundo paso consiste en acceder al área del currículo para añadir nuevas publicaciones, en la que solo hace falta una simple petición *GET* a la url correcta para lograrlo.En la figura (5.10) podemos ver que estructura tiene esta petición.

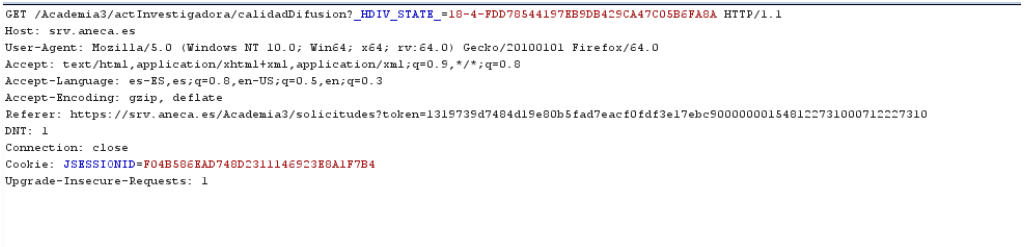

Figura 5.10: Análisis de la peticion *GET* para acceder al *CV*.

El tercer paso y el más difícil pues en el parámetro "data"de la petición *POST* para enviar los datos, se incluye una pequeña cadena de texto o *token* creada aleatoriamente para entorpecer las labores de automatización. Sin embargo esta cadena solamente se regenera cuando se realiza una petición *GET* para subir un tipo de publicación , esto quiere decir que para todos los artículos indexados podemos usar el mismo *token* extraído de la respuesta a la petición *GET* del servidor y así con todos los tipos de publicaciones.(5.11)

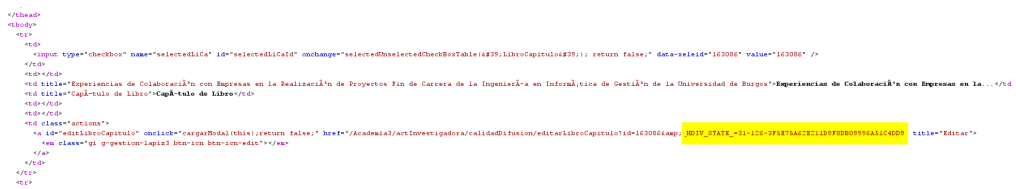

Figura 5.11: Análisis de la respuesta *HTML* a la petición *GET* para añadir publicaciones, en subrayado se encuentra el *token*.

Así pues la solución pasa por agrupar todas las publicaciones según su tipo realizar la petición *GET* para poder obtener el *token* y realizar tantas peticiones *POST* , como publicaciones haya en la lista, y repetir este proceso con cada tipo destino de publicación. Burlando así el sistema de entorpecimiento de la automatización.

Trabajos relacionados

En cuanto al apartado de la extracción de datos provenientes de las páginas ya mencionadas, lo más parecido que se podría encontrar serían las propias librerías ya usadas para este proyecto, así como algunas otras de funcionalidad muy similar

6.1. Google Scholar

scholar.py [4] Módulo para *Python* que permite realizar peticiones y analiza las respuestas para devolver únicamente la información relevante

6.2. Scopus

scopus[18] Módulo de *Python* que interactúa con la propia *API* de *Scopus* para analizar los datos extraídos de las peticiones y devolver los datos relevantes con una estructura *pandas DataFrame*[5]

6.3. Web of Science

wos [7] Herramienta que combina *Python* y scripts en *bash* para acceder a *Web of science* y extraer la información.

En cuanto al proceso de subida de los datos a la aplicación *ACADEMIA*, tras una breve búsqueda por Internet , podemos encontrar que no existe ninguna aplicación o script que realice esta labor, al menos de manera pública. Puede darse el caso de que algún investigador haya desarrollado

algún tipo de script para realizar una labor similar al propósito de este proyecto, pero con carácter personal y privado.

Como vemos no existe nada que se parezca en su totalidad a la idea de este proyecto, que combine las dos ideas fundamentales de extraer y enviar.

Conclusiones y Líneas de trabajo futuras

En este apartado vamos a exponer las conclusiones extraídas de la realización de este proyecto y las posibles líneas futuras de desarrollo para la continuidad del proyecto.

7.1. Conclusiones

Una vez finalizado el proyecto podemos decir que:

- El objetivo en líneas generales del proyecto se ha cumplido satisfactoriamente, se ha conseguido crear una herramienta funcional que permita a los investigadores recolectar y actualizar la información referente a su curricular de una manera sencilla e intuitiva.
- Ha sido satisfactorio comprobar que las técnicas y conocimientos aprendidos a lo largo de la carrera han sido útiles.
- Por otra parte, otro de los objetivos de este proyecto era adquirir nuevos conocimientos y técnicas, objetivo que considero satisfactoriamente cumplidos. Se ha profundizado en las técnicas de *Web Scraping*, tratado de datos bibliográficos y la planificación y documentación de proyectos.
- Gracias a los distintos problemas encontrados durante el desarrollo, se ha podido aprender acerca del tratamiento de los imprevistos y los problemas, así como en la búsqueda de soluciones y alternativas. Conceptos, aunque quizás no tan relacionados con el grado en sí, se entiende que son valiosos y necesarios para el desarrollo de una carrera profesional.

7.2. Líneas de trabajo futuras

Cabe aclarar que, aunque se proceda a la entrega del Trabajo Fin de Grado (TFG), esto no quiere decir que sea un proyecto totalmente perfecto y cerrado, sino que hay ciertos aspectos que se pueden mejorar para mejorar la funcionalidad y la experiencia del usuario:

- Dotar a la aplicación de una interfaz renovada, mas *"moderna"* mediante el uso de un fichero CSS que vaya más allá de las limitaciones que tiene la librería usada (Tkinter), la cual proporciona herramientas suficientes para ser funcional, pero queda algo anticuada con respecto a otras aplicaciones de uso cotidiano.
- Añadir una funcionalidad que permita, no solo subir la información bibliográfica sino también el propio texto de la aplicación, como forma de aportar más datos e información para el proceso de acreditación y evaluación.
- Mejorar el número de datos que tiene actualmente el objeto de *Python* que contiene los datos referentes al índice de impacto de las distintas revistas, aumentando en todo lo posible el número de revistas de las que posee la información para que el número de consultas a la *JCR InCites* sea el mínimo posible , haciendo que el proceso más ágil.

Bibliografía

[1] Academia. Aplicación informática, 2018. URL https://sede.educacion.gob.es/sede/login/inicio.jjsp?idConvocatoria=590.

[2] Academia. Sede electrónica del mecd, 2018. URL http://www.mecd.gob.es/redirigeme/?ruta=/servicios-al-ciudadano-mecd/catalogo/general/educacion/academia/ficha/academia.html. asd.

[3] Academia. Programa de evaluación, 2019. URL http://www.aneca.es/Programas-de-evaluacion/Evaluacion-de-profesorado/ACADEMIA.

[4] ckreibich. Scholar.py github repository, 2019. URL https://github.com/ckreibich/scholar.py.

[5] Pandas DataFrame. Documentation, 2019. URL https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.DataFrame.html.

[6] Elsevier. Scopus, 2019. URL https://www.scopus.com/home.uri.

[7] enricobacis. wos github repository, 2019. URL https://github.com/enricobacis/wos.

[8] Regular expressions operations. Python documentation, 2019. URL https://docs.python.org/3/library/re.html.

[9] Google. Scholar, 2019. URL https://scholar.google.es.

[10] ISI. Web of science, 2019. URL https://login.webofknowledge.com.

[11] JCR. Incites, 2019. URL https://jcr.incites.thomsonreuters.com/JCRJournalHomeAction.action.

[12] JCR. Incites, 2019. URL http://jcr.incites.thomsonreuters.com/JCRJournalProfileAction.action?pg=JRNLPROF&journalImpactFactor=n%2Fa&year=2017&journalTitle=COMPUT%20J&edition=SCIE&journal=COMPUT%20J.

[13] Jupyter. Notebook start guide, 2019. URL https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html.

[14] OrganicIrradiation. Scholarly github repository, 2019. URL https://github.com/OrganicIrradiation/scholarly.

[15] Pickle. Python documentation, 2019. URL https://docs.python.org/3/library/pickle.html.

[16] Requests. Python documentation, 2019. URL http://docs.python-requests.org/en/master/.

[17] sciunto org. Bibtexparser github repository, 2019. URL https://github.com/sciunto-org/python-bibtexparser.

[18] scopus api. Scopus github repository, 2019. URL https://github.com/scopus-api/scopus.

[19] Scrum. Methodology, 2019. URL http://scrummethodology.com/.

[20] Selenium. Webdriver documentation, 2019. URL https://www.seleniumhq.org/docs/01_introducing_selenium.jsp.

[21] Sublime. Text-3, 2019. URL http://www.sublimetext.com/.

[22] Tkinter. Python documentation, 2019. URL https://docs.python.org/2/library/tkinter.html.

[23] Unittest. Python documentation, 2019. URL https://docs.python.org/2/library/unittest.html.

[24] Wikipedia. Bibtex — wikipedia, la enciclopedia libre, 2019. URL https://es.wikipedia.org/wiki/BibTeX.

[25] Wikipedia. Bibtex — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/BibTeX.

[26] Wikipedia. Coockie — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/HTTP_cookie.

[27] Wikipedia. Endnote — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/EndNote.

[28] Wikipedia. Github — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/GitHub.

[29] Wikipedia. Http — wikipedia, la enciclopedia libre, 2019. URL https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto.

[30] Wikipedia. pip — wikipedia, la enciclopedia libre, 2019. URL https://es.wikipedia.org/wiki/Pip_(administrador_de_paquetes).

[31] Wikipedia. Pycharm — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/PyCharm.

[32] Wikipedia. Http — wikipedia, la enciclopedia libre, 2019. URL https://es.wikipedia.org/wiki/Python.

[33] Wikipedia. Ris — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/RIS_(file_format).

[34] Wikipedia. Session — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/Session_(computer_science).

[35] Wikipedia. Texmaker — wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/wiki/Texmaker.

[36] Wikipedia. Web scraping — wikipedia, la enciclopedia libre, 2019. URL https://es.wikipedia.org/wiki/Web_scraping.

[37] Virginia Williams. Computer science 111 fundamentals of programming i user interfaces introduction to gui programming, 2019. URL https://slideplayer.com/slide/6213253/.

[38] zhiyzuo. Python-scopus github repository, 2019. URL https://github.com/zhiyzuo/python-scopus.