UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Análisis de rendimiento de autores científicos

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Cardona Maroto, Iván

Tutores: Martínez-Plumed, Fernando

Ferri Ramírez, César

2018/2019

Agradecimientos

Me gustaría agradecer y dedicar este proyecto a todas las personas que me han ayudado directa o indirectamente en el trascurso de esta etapa en la universidad como en el desarrollo de este proyecto.

Principalmente a mi familia, la cual ha permitido que yo pueda tener esta magnífica oportunidad de estudiar una carrera universitaria y siempre me ha apoyado y animado en las decisiones que he tomado en mi andadura hasta aquí.

También quería dedicar unas palabras a todo mi círculo de amigos que ha estado partícipe en mi carrera hasta este punto. No habría sido posible llegar hasta aquí sin sus ánimos y sin su ayuda.

Obviamente, quería agradecer también a mis dos tutores, Cèsar Ferri Ramírez y Fernando Martínez Plumed, el hecho de orientarme y echarme una mano en los momentos oportunos a lo largo del desarrollo de este proyecto.

En definitiva, muchas gracias a todas aquellas personas que me ofrecieron cualquier tipo de ayuda o apoyo, sin ellas no hubiera sido posible llegar donde me encuentro actualmente.

Resumen

La extracción de datos bibliográficos como publicaciones científicas provenientes de distintos orígenes de datos, y el posterior análisis de los resultados obtenidos, es un método muy utilizado por investigadores en universidades y empresas de todo el mundo cuando estos necesitan identificar temas de interés, autores, universidades y países, en base a determinados criterios. En este sentido han aparecido métodos bibliométricos que han permitido determinar y diseñar métricas (e.g., indicadores científicos) objetivos que permiten medir y evaluar la popularidad y el impacto de artículos, autores y publicaciones específicas.

En este trabajo se pretende desarrollar una aproximación para el análisis bibliométrico de la producción científica por autores relacionados con el área de las ciencias de la computación. Para ello se han utilizado técnicas de web-scraping que, a partir distintos orígenes de datos, nos ha permitido obtener tanto la producción científica como la repercusión de los distintos documentos publicados. Mediante este desarrollo, un usuario podrá, a partir del nombre de un autor científico, obtener un listado de todas sus publicaciones, así como una descripción bibliométrica con características más relevantes de dichas publicaciones, incluyendo sus medidas de impacto, rankings, citas, etc. La dificultad radica en que, actualmente, toda esta información se encuentra en bases de datos y repositorios heterogéneos en diferentes orígenes y gestionados por distintas organizaciones. Por tanto, la solución aquí presentada constituye un primer paso para caracterizar de forma centralizada y estructurada la producción científica de autores, lo que permitirá orientar a los usuarios a determinar el nivel o prestigio tanto del autor como de sus producciones científicas.

Palabras clave: Indicadores científicos; Análisis bibliométrico; Web-Scraping; Python; Google Scholar; DBLP.

Abstract

The extraction of bibliographic data such as scientific publications from different data sources, and the posterior analysis of the results, is a commonly used by researchers in universities and companies whenever they need to identify leading topics, authors, universities and countries, according to certain criteria. In this regard, bibliometric methods have appear aiming at determining and devising metrics (e.g., scientific indicators) as objective as possible to determine the popularity and impact of specific articles, authors, and publications.

The aim of this work is to develop an approach for the bibliometric analysis of scientific production (author-wise) related to the area of computer science. For this purpose, web-scraping techniques have been applied to different data sources allowing us to obtain both the scientific production and the repercussion of the different published documents. Through this development, users will be able, from the name of a scientific author, to obtain a list of all their publications, as well as a bibliometric description with the most relevant characteristics of these publications, including their impact measurements, rankings, citations, etc. The difficulty lies in the fact that, at present, all this information is stored and provided in heterogeneous databases and repositories from different origins and managed by different organizations. In this regard, the solution presented here constitutes a first step to characterize, in a centralized and structured way, the scientific production of authors, which will guide users to determine the level or prestige of both the author and their scientific productions.

Keywords: Scientific indicators; Bibliometric analysis; Web scrapping; Python; Google Scholar; DBLP.

Tabla de contenidos

Índice de figuras

1. Introducción

1.1. Motivación

En la actualidad, la toma de decisiones en política científica constituye un proceso muy relevante, pero a su vez complejo. Para facilitar este proceso, parte de la ciencia ha dedicado tiempo a idear métricas objetivas que puedan orientar al ser humano a la hora de tomar decisiones o hacerse una idea del elemento calificado. No obstante, la integración de estos sistemas de indicadores científicos tampoco resulta una tarea fácil. Estos indicadores científicos pueden abarcar distintas ramas y dimensiones de la ciencia.

Observando la rama en la que va centrada nuestro proyecto, la literatura científica, observamos la existencia de indicadores científicos utilizados frecuentemente. Por ejemplo, existen indicadores que aportan una calificación atendiendo al número de citas que recibe un artículo de un determinado autor, existen ciertas métricas cuyo objetivo es calcular el nivel de ciertas revistas o conferencias, etc. Todos estos indicadores reúnen un fin, y es el de aportar al ser humano una calificación calculada atendiendo a los parámetros que se creen convenientes para que tenga una visión del nivel del elemento evaluado.

La principal motivación del proyecto es ofrecer al usuario una aproximación (en forma de portal web) que ofrezca y reúna de forma estructurada y completa los datos más característicos de la producción científica de un determinado autor, así como los indicadores bibliométricos de producción, circulación, dispersión y visibilidad de la misma, incluyendo las métricas más relevantes para la evaluación de conferencia y/o revistas científicas. De esta forma y haciendo uso de estos indicadores poder orientar al usuario para que saque las conclusiones sobre el nivel que muestren dichas calificaciones. Así mismo, poder recoger y unificar en un mismo portal web las distintas métricas para evaluar conferencias y revistas.

1.2. Objetivo del proyecto

El objetivo general del proyecto es desarrollar una solución sencilla y amigable (e.g., aplicación web) que facilite los datos de las publicaciones de un autor, así como las distintas métricas empleadas, con el fin de que el usuario pueda realizar un análisis de la

publicación. Para llegar a dicho objetivo hemos detallado una serie de tareas específicas, estas son:

- Análisis de los datos de las plataformas donde se llevará la extracción de datos.
- Extracción de los datos seleccionados mediante técnicas de *web scraping*.
- Extracción de los indicadores científicos.
- Desarrollo del entorno web.
- Implementación de la funcionalidad a la web.

1.3. Estructura de la memoria

Tras esta primera introducción, vamos a pasar al segundo apartado de este proyecto. En este segundo capítulo se analiza el estado del arte del proyecto, donde se detallan las distintas plataformas empleadas en este proyecto, así como otros ejemplos similares a éstas. Se explican también conceptos claves para entender en síntesis todo el objetivo de este proyecto.

La tercera parte de este proyecto va dedicada a la explicación detallada de los datos tratados en este proyecto. Desde los datos extraídos de las plataformas, hasta las distintas métricas bibliográficas empleadas.

Seguidamente, en el cuarto capítulo, se detallan las herramientas y tecnologías de las que nos hemos servido para desarrollar el proyecto. En esta parte se explican el software para el desarrollo empleado, los lenguajes de programación, las tecnologías empleadas y algunas de las librerías más importantes para el desarrollo.

En el quinto punto se explica en mayor profundidad el proceso de desarrollo que se ha seguido para llegar a la solución. Relacionado con el quinto punto está el punto seis, el cual argumenta la implantación que se ha llevado a cabo y muestra los resultados obtenidos, explicando detalladamente el funcionamiento de la aplicación con ejemplos concretos.

Finalmente, en el séptimo punto se encuentran las conclusiones obtenidas respecto a la solución del proyecto y se aportan sugerencias que se podrían implementar en un trabajo futuro para la mejora de dicha aplicación.

2. Bibliometría

2.1. Marco teórico

Antes de comenzar con la explicación y desarrollo del proyecto consideramos conveniente introducir una serie de términos que pensamos que son clave y que son repetidos a lo largo de este proyecto:

- **Publicación**: Cuando nos referimos al término publicación, hablamos para aludir a la obra de un autor que ha sido publicada. Puede ser una revista, un libro, una tesis, etc.
- **Cita**: Se conoce el concepto cita como la forma abreviada de referenciar al lector de donde se ha extraído el conjunto de datos bibliográficos en el que nos hemos apoyado para presentar nuestro trabajo.
- **Indicador científico**: Los indicadores científicos se tratan de métricas basadas en mediciones realizadas de forma objetiva con el fin de servir de ayuda en la toma de decisiones dentro de las distintas dimensiones de la ciencia. En definitiva, se tratan de métricas que ayudan al humano a generar una idea del nivel que tiene el elemento evaluado.
- **API**: Se tratan de las siglas de '*Application Proggraming Interface*' y consiste en un conjunto de código que sirve como interfaz permitiendo la comunicación entre varios programas diferentes.
- **Bases de datos**: Se basa en un sistema de archivos electrónico donde se reúnen los distintos tipos de datos que se requieran almacenar. Se puede considerar también como un conjunto de información organizada.
- **Motor de búsqueda**: Conocido también como buscador, es un sistema informático cuya función es buscar archivos almacenados en servidores web. La búsqueda se inicia con las palabras clave que introduce el usuario y el resultado es una serie de direcciones web que tienen relación con la palabra clave introducida.

2.2. Bibliometría

Hoy en día, internet se ha convertido en una herramienta fundamental, la cual puede verse como una fuente de información a la que recurrir, como lugar entretenimiento, como portal de comunicación, etc. Este uso continuado ha supuesto que la cantidad de datos existentes en la red haya ido aumentando desde sus inicios hasta el día de hoy. El uso de internet como fuente de información ha producido que se almacenen en la red una cantidad inmensa de documentos de todo tipo [25].

La ciencia es una actividad intelectual que se caracteriza por tener como finalidad intentar dar siempre una respuesta o razonamiento a preguntas desde una perspectiva empírica. La ciencia lleva desde tiempos inmemorables entre la sociedad, sin embargo, no es hasta hace relativamente poco que se comenzó a investigar y a analizar su naturaleza. Por ejemplo, la medida de magnitudes como el número de publicaciones científicas, la medición de impacto, etc. Esta rama de la ciencia de la que acabamos de hablar se conoce como cienciometría [2].

Puede considerarse que forma parte de la cienciometría el concepto conocido como bibliometría. La bibliometría consiste en el cálculo y análisis de aquello que se puede cuantificar dentro del ámbito de la producción científica. Esta serie de magnitudes de las que hablamos, reciben el nombre de indicadores bibliométricos. En nuestro proyecto, hemos recopilado una serie de indicadores bibliométricos, han sido extraídos de diferentes plataformas que promueven esta ciencia, la bibliometría [2].

2.2.1. Índices de impacto de conferencias

El origen de los datos sobre las conferencias viene dado por *The GGS Conference Rating*. Este proyecto es una iniciativa del grupo conocido como GGS: **GII-GRIN-SCIE** [22]. GII es un grupo formado por una serie de profesores italianos de ingeniería informática (Gruppo di Ingegneria Informatica), los cuales imparten clases en la universidad politécnica de Ancona [13]. De la misma forma, las siglas GRIN (GRuppo di INformatica) representan a otro grupo de profesores italianos de ingeniería informática, los cuales tienen actividad docente en diferentes universidades de Italia (Pisa, Génova, Milán, Roma y Bolonia) [12]. Por otro lado, las siglas SCIE representan a la Sociedad Científica Informática de España [21].

La web de esta asociación ofrece la posibilidad de descargar el documento donde se almacenan las distintas calificaciones puntúa a las conferencias. De este documento del cual nos hemos servido, hemos recogido las siguientes columnas: *GGS Rating*, *GGS Class* y *Qualified Classes* [22].

- **GGS Rating**: Este atributo representa la puntuación que se le asocia a la conferencia. Viene calculada por las distintas puntuaciones recibidas por los distintos factores o clases. La nota sigue la siguiente escala en orden decreciente: **A++, A+, A, A-, B+, B-, C**.
- **GGS Class**: Se podría decir que el valor de este campo representa el "nivel" de la conferencia, el cual va íntimamente ligado a la puntuación.

GGS Class	GGS Rating	Descripción
1	A++, A+	Conferencias de alto nivel
2	A, A-	Eventos de alta calidad
3	B, B-	Eventos de buena calidad
-	Work in progress	En proceso de puntuar

La tabla que se muestra justo arriba muestra la relación que hay entre la puntuación establecida con el nivel de la conferencia [22].

- **Qualified Classes**: este atributo está constituido por las notas que exponen los diferentes factores o clases, como son denominadas en la web. El algoritmo se basa en tres clases diferentes: CORE, MA y LiveSHINE.

  - **CORE**: Las siglas corresponden a 'the *Computing Research and Education Association of Australia'*. El nacimiento de esta puntación se debe a la fuerte experiencia que tienen en Australia puntuando conferencias, es por ello, que en 2008 idearon un sistema de puntuación propio. Actualmente, se tienen en cuenta las puntuaciones de CORE 2018 [7].

Puntuación	Descripción
A+	Conferencias líderes en un campo
A	Conferencias respetadas en un campo
B	Buenas conferencias
C	Conferencias que cumplen los estándares mínimos
L	Conferencias locales de Australia

- **MA**: *Microsoft Academic* se trata de una sección de la API de *Microsoft Knowledge*. Esta API aporta indicadores bibliométricos sobre conferencias de carácter informático.
- **LiveSHINE**: Este factor es el sucesor de factor conocido como SHINE, basado en la puntuación de conferencias de Google Scholar. Hoy en día, LiveSHINE se trata de una extensión de Google Chrome donde se puede consultar el *H-Index*[1] de conferencias de carácter informático.

Para mantener un estándar de puntuaciones, tanto para la medición de MA, como para la de LiveShine, GGS ha establecido una equivalencia atendiendo al valor que devuelve *H-Index*.

Puntuación de H-Index	GGS Rating
1 – 50	A++
51 – 75	A+
76 – 200	A
201 – 250	A-
251 – 575	B
576 - 650	B-
Demás resultados	C

Esta asociación ha perfeccionado esta herramienta para calificar las conferencias, para ello, han desarrollado un algoritmo el cual basa sus puntuaciones en las métricas anteriormente detalladas. Las calificaciones de este algoritmo pueden observarse en la Tabla 1 del Anexo.

2.2.2. Índices de impacto de revistas

Para la extracción de las puntuaciones de las revistas hemos recurrido al conjunto de datos obtenido de la plataforma Scopus, el cual recopila información adicional sobre la revista (si está en activo o no, el año cuando se inició dicha revista, etc.) y las distintas métricas que hemos recopilado. Dicho documento guarda las puntuaciones de 2015, 2016 y 2017 y procede del portal de *Scopus* el cual hemos explicado antes brevemente. Los factores de los que estamos hablando son CiteScore, SJR y SNIP.

[1] El *H_Index* es un sistema de medida promovido por George Hirsch para medir en nivel de calidad del autor a partir de la cantidad de citas que han recibido sus artículos.

- **CiteScore**: Esta métrica representa la relación existente entre el número de citas que recibe los artículos de un autor por artículo publicado. Como las demás métricas de las hemos hablado, su propósito es evaluar revistas. El cálculo de esta métrica se basa en la división del número de citas de todo un año entre el número de publicaciones de los tres años anteriores [19]. Por ejemplo:

$$CiteScore\ 2018 = \frac{Citas\ 2018}{Número\ publicaciones\ 2015, 2016\ y\ 2017}$$

- **SJR**: Este factor también es conocido como SCImago Journal & Country Rank, fue desarrollado por el grupo Scimago. La característica principal de esta métrica es que a la hora del cálculo basa su resultado en el número de citas totales obtenidas, otorgando un valor a la cita atendiendo al prestigio de la revista de la que proviene la cita. Cuanto más prestigio tenga la revista que cita, más valor tendrá esa cita a la hora de calcular esta métrica. A continuación, se explicará la fórmula utilizada para obtener el resultado de esta métrica [14].

  - **Cálculo de la métrica SJR:**

    En el proceso del cálculo se distinguen dos fases, una primera donde se calcula el prestigio de la revista (**PSJR2**), y una segunda donde se halla el valor de la métrica (**SJR2**).

    FASE 1: Cálculo del prestigio

    En una primera instancia, el prestigio de cada revista tiene el mismo valor: 1/*N*, donde N representa el número total de revistas almacenadas en la base de datos. Seguidamente, empieza un proceso iterativo que basa su cálculo en tres criterios [14]:

    1. Prestigio mínimo: Otorgan un valor mínimo de prestigio únicamente por estar almacenada la revista en la base de datos.
    2. Prestigio de revistas: Otorgan un valor de prestigio atendiendo al número de documentos almacenados en la base de datos.
    3. Prestigio de citas: Valor de prestigio calculado debido al número de citas, a la importancia de las citas y a la cercanía de las citas.

$$PSJR2_i = \overbrace{\frac{(1-d-e)}{N}}^{1} + \overbrace{e \cdot \frac{Art_i}{\sum_{j=1}^{N} Art_j}}^{2} + \overbrace{\frac{d}{PSJR2D} \cdot \left[\sum_{j=1}^{N} Coef_{ji} \cdot PSJR2_j\right]}^{3}$$

$$Coef_{ji} = \frac{(Cos_{ji} \cdot C_{ji})}{\sum_{h=1}^{N}(Cos_{jh} \cdot C_{jh})}$$

$$PSJR2D = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{(Cos_{ji} \cdot C_{ji})}{\sum_{h=1}^{N}(Cos_{jh} \cdot C_{jh})} \cdot PSJR2_j$$

Tabla de variables [14]:

Variable	Descripción
N	Número de revistas en la base de datos
C_{ji}	Referencias de la revista *j* a la revista *i*
d	Constante. $d = 0{,}9$
e	Constante. $e = 0{,}0999$
Art_j	Número de documentos citados en *j*
Cos_{ji}	Coseno entre las co-citas[2] de *j* e *i*

FASE 2: Cálculo de la métrica SJR

Una vez calculado el prestigio de la revista, se procede a determinar el valor del factor SJR, y la fórmula utilizada para ello es la siguiente:

$$SJR2_i = \frac{PSJR2_i}{\left(Art_i / \sum_{j=1}^{N} Art_j\right)} = \frac{PSJR2_i}{Art_i} \cdot \sum_{j=1}^{N} Art_j$$

- **SNIP**: SNIP (*Source Normalized Impact per Paper*) se trata de una métrica que se apoya en la comparación de publicaciones dentro de sus campos temáticos, teniendo en cuenta el número de veces que los autores citan otros documentos, así como el

[2] El término *co-cita* es usado para designar a la frecuencia con la que dos documentos *i* y *j* son citados conjuntamente por otros documentos.

impacto de la cita. Su cálculo consiste en la división del número medio de citas recibido por los artículos de una revista a lo largo de tres años (*Raw Impact per Paper*, **RIP**) entre la citación potencial del campo científico de la revista (*Relative Database Citation Potencial*, **RDCP**) [19].

$$SNIP = \frac{RIP}{RDCP}$$

3. Herramientas de la bibliometría

Así como la cantidad de documentos ha ido aumentando, el número de portales web donde son almacenados o el número de buscadores ha aumentado también. Es en la rama de bases de datos documentales y motores de búsqueda que faciliten el acceso a documentos donde centraremos parte de nuestro estudio.

Un claro ejemplo sobre base de datos bibliográfica es la plataforma *Scopus*, la cual es una base de datos sobre ciencia y tecnología (química, física, medicina, biología, ingeniería, ciencias sociales, etc.). El uso de esta herramienta permite el acceso a publicaciones científicas y sus numerosas referencias bibliográficas [5]. Según la web oficial de *Scopus*, el almacenamiento de su base de datos alcanza a albergar más de 24.000 títulos provenientes de más de 5.000 editores internacionales [19].

Figura 1: Logotipo de Scopus.

Además de ofrecer una gran cantidad de documentos y su respectiva información, también ofrece diversas métricas que pueden dar una orientación sobre la calidad de la revista donde han sido publicados dichos documentos. Estas métricas se conocen también por el nombre de indicadores científicos.

Por otro lado, como ejemplo de motor de búsqueda de literatura científica se puede hablar de otra plataforma llamada *CiteSeer*^{x}. Es un buen ejemplo debido a que, según los datos que aporta la web oficial del buscador: "*CiteSeer*^{x} fue la primera biblioteca digital y motor de búsqueda en proporcionar indexación de citas y enlaces hacia las citas de manera autónoma.". Esta herramienta, fue desarrollada en 1997 y en sus inicios llegaba a atender 1.5 millones de solicitudes diarias por parte de los usuarios llegando a superar las capacidades del propio sistema, según los datos de la web [4].

Figura 2: Logotipo de CiteSeer^{x}

En nuestro proyecto, recopilaremos la información de este tipo de sitios, tanto de bases de datos digitales, como de motores de búsqueda, concretamente, DBLP y Google Scholar, respectivamente.

3.1. Google Scholar

Google Scholar es un motor de búsqueda académico desarrollado por Google dedicado a recoger documentos científicos y las citas que estos han recibido. Esta herramienta fue lanzada en 2004 casi al mismo tiempo que *Scopus*, la herramienta detallada anteriormente. No obstante, un detalle importante a destacar es que, aunque salieran a la luz casi al mismo tiempo y la finalidad fuera parecida (ofrecer una fuente información científica a los usuarios), tenían un enfoque distinto. Mientras *Google Scholar* fue desarrollada como una herramienta dinámica, donde todo fuera completamente automático, es decir, el motor de búsqueda mostraría los resultados encontrados por su algoritmo de búsqueda, *Scopus* fue concebida como una herramienta más hermética, donde el humano sí que tendría que intervenir para tener más control sobre los datos almacenados [15].

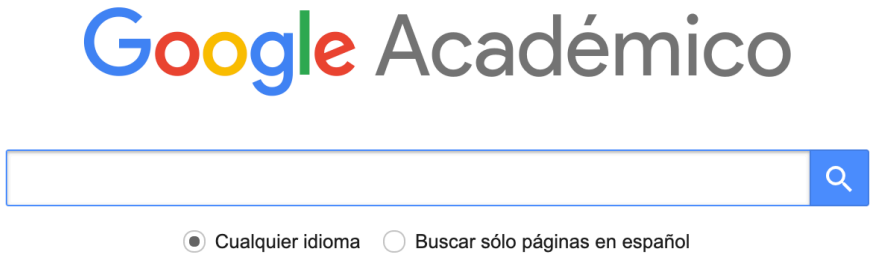


Figura 3: Interfaz de búsqueda de Google Scholar

Como se aprecia en la figura 3, la interfaz de *GS* ofrecida al usuario para realizar una búsqueda es muy parecida a la ofrecida por el buscador Google, lo cual hace que no parezca complejo para el usuario el uso de esta herramienta.

El motor de búsqueda utilizado por *Google Scholar* funciona de tal manera que genera un rastreo por la red sobre distintos dominios institucionales de libre acceso pertenecientes a universidades, repositorios, páginas web de revistas, bases de datos documentales o catálogos de bibliotecas. Una vez termina el rastreo y recopilados los documentos, dichos documentos son indexados registrando toda su información bibliográfica incluyendo las citas bibliográficas. Los resultados de dicho rastreo suelen tener un repertorio de formatos: *doc*, *ppt*, *html*, *pdf* (en la mayoría) o incluso en ocasiones *postScript* [23].

A la hora de buscar se puede recurrir a operadores que mejoran y facilitan la búsqueda al usuario. Por ejemplo:

- '+': Utilizado para incluir palabras vacías.
- '**-**': Utilizado para excluir palabras de la búsqueda.
- '**OR**': Utilizado para expandir los resultados posibles de la búsqueda.
- '**filetype**': Utilizado para especificar el formato que se desea de la búsqueda.
- **Uso de comillas** para determinar palabras o frases exactas.

Figura 4: Resultado que nos dirige a la fuente del documento

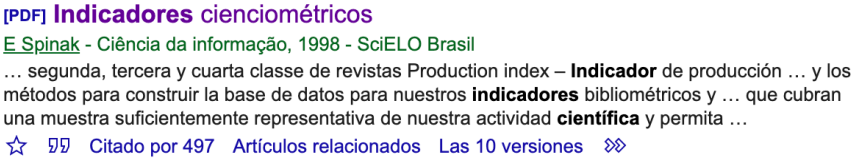


Figura 5: Resultado que nos dirige al documento original

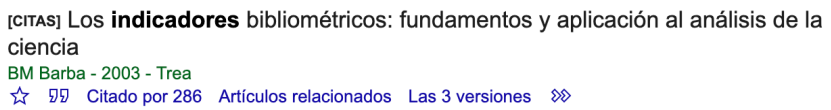


Figura 6: Resultado que nos devuelve una cita

En 2012, Google lanzó dos herramientas complementarias a Google Scholar: *Google Scholar Citations* y *Google Scholar Metrics*. La primera, *Google Scholar Citations*,

ofrece al usuario poder crearse una cuenta como investigador y poder averiguar el índice-H. Por otro lado, *Google Scholar Metrics* tiene un propósito parecido, pero a nivel global. Permite conocer el índice-H de una gran cantidad de revistas y más fuentes documentales.

Para resumir la capacidad que tiene esta plataforma podemos enumerar una serie de características que consideramos importantes y que hacen que este al nivel de otras plataformas o incluso destaque [23].

- Se trata de una plataforma de acceso libre.
- Ofrece en sus resultados acceso directo a publicaciones.
- Los resultados muestran distintos tipos de documentos: artículos de revistas, tesis, libros, informes, etc.
- Consigue extraer las distintas versiones que tiene una publicación.
- Gracias a las herramientas *Google Scholar Citations* y *Google Scholar Metrics* permite analizar fuentes documentales gracias al índice-H.
- Cobertura de documentos en lenguas nacionales europeas.

Finalmente, cabe destacar que Google Scholar no tiene oficialmente una API como otras plataformas, sin embargo, han sido desarrolladas varias versiones por usuarios con el fin de tener una opción a la hora de extraer datos de dicha plataforma.

Para poder realizar dicha acción en nuestro proyecto hemos recurrido a dos APIs, la proporcionada por DBLP y una creada por un usuario que la hizo pública en la plataforma GitHub para extraer información con Google Scholar (https://github.com/ckreibich/scholar.py).

La API *scholar.py* se trata de un módulo programado en Python cuyo funcionamiento está dedicado para realizar consultas vía terminal, no obstante, las clases implementadas pueden tener un uso independiente. Para la extracción de datos esta API se basa fundamentalmente en la conocida librería *Beautiful Shop* [1]. Algunos rasgos que ofrece dicha API son:

- Puede extraer el número total de resultados reportados por Google Scholar.
- Puede extraer los datos más relevantes de una publicación (título, número de citas, número de versiones, enlace del PDF, …).
- Admite recuperar información de las citas en formatos no proporcionados por Google Scholar, por ejemplo, *BibTex* o *EndNote*.

- Imprime por terminal en formato CSV, en simple texto o en el formato especificado de la exportación de citas.

```
$ scholar.py -c 1 --author "albert einstein" --phrase "quantum theory"
         Title On the quantum theory of radiation
           URL http://icole.mut-es.ac.ir/downloads/Sci_Sec/W1/Einstein%201917.pdf
          Year 1917
     Citations 184
      Versions 3
    Cluster ID 17749203648027613321
      PDF link http://icole.mut-es.ac.ir/downloads/Sci_Sec/W1/Einstein%201917.pdf
Citations list http://scholar.google.com/scholar?cites=17749203648027613321&as_sdt=2005&sciodt=0,5&hl=en
 Versions list http://scholar.google.com/scholar?cluster=17749203648027613321&hl=en&as_sdt=0,5
       Excerpt The formal similarity between the chromatic distribution curve for thermal radiation [...]
```

Figura 7: Ejemplo de consulta que realiza la API scholar.py [1].

Es con esta API de la cual nos servimos para extraer los datos que creemos más relevantes, los datos obtenidos son los siguientes:

- **Título de la publicación**: Es el título dado por el autor al artículo.
- **URL**: Enlace al apartado web que nos dirige a la fuente del artículo.
- **Año**: Año cuando fue publicado el artículo
- **Número de citas**: Cantidad de veces que registra Google Scholar que ha sido citado el artículo.
- **Número de versiones**: Cantidad de versiones que tiene el documento.
- ***Cluster id***: Se trata de un número identificador del *cluster* donde es guardado el artículo a buscar.
- **URL del PDF**: Enlace donde se puede visualizar el documento PDF.
- **URL de las versiones**: Enlace que muestra los enlaces a las distintas versiones que ha tenido el artículo buscado.
- **URL de las citas**: Enlace en el cual se muestran los diferentes enlaces a las citas que ha tenido el documento.
- **Resumen**: Es el resumen del autor al artículo buscado en cuestión.

3.2. DBLP

DBLP es un sitio web que funciona como bibliografía informática creado en 1993 por la Universidad de Trier, Alemania. Originalmente, las siglas DBLP significaban *Database and Logic Programming*, no obstante, hoy en día han cambiado de significado empleando el nombre de *Digital Bibliography & Library Project*. Empezó formándose a

partir de una pequeña colección de archivos HTML, desde entonces hasta la actualidad ha ido creciendo año tras año. En junio de 2019 la propia plataforma detalló que la base de datos almacenaba alrededor de 4.4 millones de documentos que habían sido publicados por más de 2.2 millones de autores [8].

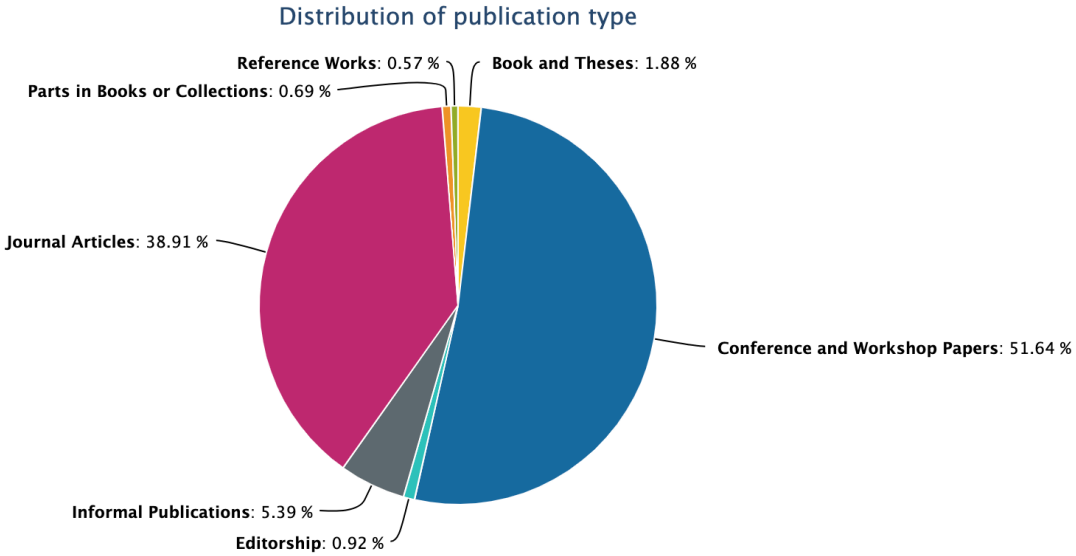


Figura 8: Gráfico de la distribución del tipo de publicaciones almacenadas en la BBDD de DBLP [8].

Como se aprecia en la estadística de la figura 7, se pueden observar cómo está repartido el contenido de la base de datos de la plataforma. La base de datos contiene:

- 2.452.748 publicaciones extraídas de conferencias (51.64%).
- 1.848.212 artículos de revistas (38.91%).
- 255.862 publicaciones informales (5.39%).
- 89.150 libros y tesis (1.88%).
- 43.810 redacciones (0.92%).
- 32.748 partes de libros y colecciones (0.69%).
- 26.996 estudios de referencia (0.57%).

La búsqueda que ofrece la plataforma también tiene diversos operadores para que la búsqueda sea más completa para el usuario [8].

- **Prefijo de búsqueda**: El usuario puede poner un prefijo en la búsqueda y la plataforma mostrará resultados que contengan dicho prefijo. Este método de búsqueda no requiere de ningún carácter específico para indicar que es un

prefijo debido a que es la forma por defecto que tiene la web para realizar una búsqueda.

- **Palabra exacta**: Para buscar una palabra específica vale con añadir al final de la palabra el signo del dólar ($). Por ejemplo, para encontrar resultados que contengan la palabra “graph” habría que especificar en la búsqueda “graph$”.
- **Booleano AND**: Si el usuario quiere buscar varias palabras que estén contenidas en un mismo resultado, deberá de poner ambas palabras separadas por un espacio sin ningún tipo de carácter especial.
- **Booleano OR**: Si en cambio el usuario requiere buscar varias palabras pero que no tienen por qué estar contenidas en un mismo resultado, deberá de especificar entre las palabras el símbolo “|”.

DBLP ofrece la posibilidad de descargar todos los datos que tienen almacenados para poder ser consultados. También ofrece una API con el fin de que los usuarios puedan extraer los datos que deseen. Dicha API da la posibilidad al usuario a buscar por publicaciones, autores o lugares donde han sido publicados los artículos. El formato de los resultados que se devuelven puede ser XML (por defecto) o JSON si lo especifica el usuario.


```
▼<hit score="6" id="4673202">
  ▼<info>
    ▼<authors>
      <author>Dale Skrien</author>
    </authors>
    ▼<title>
      A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs.
    </title>
    <venue>Journal of Graph Theory</venue>
    <volume>6</volume>
    <number>3</number>
    <pages>309-316</pages>
    <year>1982</year>
    <type>Journal Articles</type>
    <key>journals/jgt/Skrien82</key>
    <doi>10.1002/JGT.3190060307</doi>
    <ee>https://doi.org/10.1002/jgt.3190060307</ee>
    <url>https://dblp.org/rec/journals/jgt/Skrien82</url>
  </info>
  <url>URL#4673202</url>
</hit>
```


Figura 9: Resultados que ofrece DBLP.

La figura 8 se trata de un ejemplo de la estructura que presentan los resultados ofrecidos tras la utilización de la API. Además, también se pueden extraer de esa imagen los datos que, en parte, se pueden visualizar al usar la plataforma de manera manual. Es así, a través de la API que comentamos como extraemos los datos de esta plataforma. Los datos obtenidos son los siguientes:

- **Título de la publicación**: Como indica el nombre se trata del nombre representativo seleccionado por el autor o autores para el artículo.

- **Tipo de artículo**: Describe de qué tipo de publicación se trata. Puede tomar los siguientes valores [9]:
    - Book and Theses
    - Conference and Workshop Papers
    - Editorship
    - Informal Publications
    - Journal Articles
    - Parts in Book or Collections
    - Reference Works
- ***Venue***: Se trata del nombre del lugar donde ha sido publicado el artículo. Por ejemplo, el nombre de una revista o el nombre de una conferencia.

3.3. Heterogeneidad entre DBLP y Google Scholar

Ambas herramientas tienen un fin similar, ofrecer al público un lugar donde encontrar información a partir de documentos de profesionales de un sector. Sin embargo, son herramientas cuya estructura y mecanismo son totalmente distintas.

Por un lado, encontramos DBLP, la cual se trata de una herramienta cuya base es el almacenamiento de todo tipo de documentos de carácter informático, mientras que Google Scholar pone su base en buscar documentos por distintos tipos de repositorios, lo que supone que tenga ningún tipo de almacenamiento. También cabe repetir que Google Scholar abarca documentos de diferentes temas, mientras que DBLP está especialmente dedicado a documentos de carácter informático.

DBLP no ofrece el acceso directo a los documentos, solamente la información bibliográfica de éste, mientras que esta característica sí que está presente en Google Scholar. Tampoco muestra ningún tipo de indicador científico, cosa que puede resultar bastante útil para dar una idea del nivel o prestigio que puede tener un documento o un autor. Esto sí que lo tiene Google Scholar gracias a sus extensiones Google Scholar Metrics y Google Scholar Citations.

4. Tecnologías y herramientas

En el apartado actual se expondrán y se detallarán las diferentes herramientas empleadas para el desarrollo del proyecto. Primeramente, se expondrá la técnica con la cual no podríamos haber conseguido los datos, el *web scraping*, y seguidamente, se continuará con las distintas herramientas (software de desarrollo, lenguajes de programación y librerías empleadas).

Para obtener los datos de estos dos sitios (DBLP y Google Scholar), hemos recurrido a la conocida técnica de *web scraping*. El concepto de *web scraping* es conocido como el conjunto de técnicas para extraer información de páginas web de manera automática [10]. Parte del objetivo de dichas técnicas es obtener el contenido del entorno web para después estructurar ese contenido para aportar información al lector de esa información [20].

Figura 10: Proceso de la extracción de datos mediante web scraping.

Existen diversas técnicas para poder llevar a cabo la extracción de datos de una página web. Podemos pasar desde la acción manual de copiar datos y pegarlos en otro documento hasta la utilización de un software especializado para dicha extracción, donde el propio programa analiza y reconoce la estructura en la que están dispuestos los datos y los recoge. También encontramos técnicas como la petición HTTP hacía la web, la extracción de datos desde APIs, etc. [20].

Cabe destacar el software empleado para la realización de los distintos archivos del proyecto. En nuestro caso hemos utilizado Visual Studio Code, se trata de un editor de código creado por Microsoft. Hemos seleccionado este programa, principalmente, debido a los numerosos lenguajes de programación compatibles con dicho software.

Figura 11: Icono del editor Visual Studio Code

Un lenguaje de programación es la pieza base para la formación de programas. En nuestro caso hemos hecho uso del lenguaje de programación conocido como Python, concretamente, la versión 3.7.1. Fue dado a conocer en 1991 y es considerado un lenguaje multiparadigma, dedicado a la programación orientada a objetos, a la programación imperativa y a la funcional. Consta también de diversas estructuras que facilitan considerablemente las tareas con datos. Python nos ha facilitado la creación de los diversos archivos que aportan al proyecto la funcionalidad requerida [3].

Figura 12: Icono del lenguaje Python.

Otro lenguaje que ha sido usado es HTML (HyperText Markup Language), lenguaje marcado que es la base para la elaboración de páginas web y su año de lanzamiento fue en 1993. Funciona de tal forma que el navegador, donde se visualiza el fichero HTML, interpreta mediante un sistema de etiquetas la apariencia que debe de tener. Como bien se entiende, dicho lenguaje nos ha servido para desarrollar la web donde se pondrá interfaz a los resultados extraídos por los archivos de Python. Al tratarse de un lenguaje marcado, a la

hora de ejecutar acciones en el entorno web, es necesario que se apoye en otros lenguajes, en nuestro caso JavaScript.

Para poder enlazar la parte de las funciones en Python con el HTML, nos hemos apoyado en Flask. Flask es un pequeño framework programado en Python que permite el desarrollo de aplicaciones web [11].

Figura 13: Icono de Flask

Para finalizar este apartado, detallaremos las diversas librerías utilizadas a lo largo del desarrollo del proyecto:

- *BeautifulSoup*: Ofrece diversos métodos para la extracción de información de sitios web en archivos HTML o XML.
- *Urllib.request*: Consta de métodos que ofrecen la funcionalidad de poder acceder a URLs.
- *Json*: Permite codificar y decodificar datos en formato JSON.
- **Flask**:
    - *render_template*: Permite cargar una plantilla HTML.
    - *request*: Permite recoger los datos que han sido transmitidos por un método POST o PUT.
- *Xlrd*: Este módulo aporta la posibilidad de leer archivos Excel (*.xls* y *.xlsx*) con la información obtenida de las bases de datos descritas anteriormente.
- *OS*: Ofrece de manera sencilla la posibilidad de emplear la funcionalidad del sistema operativo.
- *Difflib*: Posibilita la comparación de secuencias.
- *Distance*: Compara distintas secuencias y calcula la similitud entre ambas. Se ha utilizado en este caso tres tipos de distancias: Levenshtein, Jaccard y Sorensen.

5. Implementación de la solución

En este punto del trabajo expondremos detalladamente como se ha procedido para llevar a cabo el objetivo principal del proyecto, realizar la aplicación. Primeramente, haciendo una descripción a grandes rasgos de las partes del proyecto, distinguimos dos partes clave, éstas son:

1. Desarrollo de los scripts, que devolverán los datos necesarios, en Python.
   a. Extracción de información en **DBLP**
   b. Extracción de información en **Google Scholar**
   c. Extracción de **indicadores científicos**
2. Creación del entorno web e incorporación de la funcionalidad.

5.1. Desarrollo de scripts

Para comenzar con este punto hicimos un previo repaso a las distintas plataformas a donde íbamos a realizar la búsqueda para la obtención de la información. Como se ha expuesto anteriormente, nos centramos en tres puntos: DBLP, Google Scholar y los conjuntos de datos obtenidos de Scopus y GII-GRIN-SCIE (GGS) Conference Rating, donde se reúnen los distintos indicadores científicos. La extracción de los indicadores científicos será algo distinta a la extracción de las otras dos plataformas, ya que, se accederá a dichos *datasets* a nivel local, es decir, han sido almacenados en el disco duro de la máquina que ejecuta dichas funciones.

5.1.1. Extracción de información de DBLP

Para saber cómo obtener la información de este repositorio digital, acudimos, primeramente, al apartado F.A.Q. (*Frequently Asked Questions*), apartado donde se exponen las preguntas que son hechas con suma frecuencia por parte de los usuarios. En ella, se puede apreciar una sección donde se explica la forma de utilizar la API que ofrece DBLP.

La API ofrece tres servicios, la consulta de publicaciones, la consulta de autores o la consulta de *venues*[3]. Cada consulta se realiza con una URL la cual se completa con una serie de parámetros que varían dependiendo de los resultados que se quieran obtener.

- Para publicaciones: **http://dblp.org/search/publ/api**
- Para autores: **http://dblp.org/search/author/api**
- Para *venues*: **http://dblp.org/search/venue/api**

Parameter	Description	Default	Example
q	The query string to search for, as described on a separate page.		...?q=test+search
format	The result format of the search. Recognized values are "xml", "json", and "jsonp".	xml	...?q=test&format=json
h	Maximum number of search results (hits) to return. For bandwidth reasons, this number is capped at 1000.	30	...?q=test&h=100
f	The first hit in the numbered sequence of search results (starting with 0) to return. In combination with the h parameter, this parameter can be used for pagination of search results.	0	...?q=test&h=100&f=300
c	Maximum number of completion terms (see below) to return. For bandwidth reasons, this number is capped at 1000.	10	...?q=test&c=0

Figura 14: Tabla sobre los parámetros que acepta las URL para la búsqueda [8].

En la figura 14 se puede apreciar una tabla con los parámetros aceptados por la API. Seguidamente se detallarán dichas variables:

- **q**: Es la variable que aporta el dato a buscar: el nombre de un autor, el nombre de una revista, el nombre de una publicación, etc. Cabe destacar que, en caso de buscar una cadena de palabras, estas palabras deberán estar unidas con el signo '+'.
- **format**: Es la variable que determina el formato de la respuesta, es decir, los resultados devueltos tendrán la estructura que se indique como valor. Si no se especifica se tomará una estructura XML.
- **h**: Dicho parámetro indica el máximo de resultados que dará la consulta, su valor por defecto es de 30.
- **f**: El valor asociado a esta variable indica la posición de la lista de resultados por la que empezaran a mostrarse estos. Por defecto empezará en la posición 0.
- **c**: El valor dado a este factor determina el número máximo de términos que completarán al valor dado en el parámetro 'q'. Por omisión se completará el término indicado hasta un máximo de 10 veces. Por ejemplo, al buscar la palabra "*term*" se mostrarán resultados que autocompletarán la cadena de caracteres como los siguientes: "*terms*", "*terminal*", "*termes*", etc.

[3] El concepto *venue* es un término inglés cuya traducción explícita es 'lugar' o 'sede'. En este caso, lo utilizamos para referirnos al lugar donde se publican los artículos (conferencias, revistas, etc.).

```
▼<completions total="19" computed="19" sent="10">
   <c sc="76" dc="75" oc="76" id="22746112">term</c>
   <c sc="26" dc="26" oc="26" id="22746360">terms</c>
   <c sc="16" dc="16" oc="16" id="22746186">terminal</c>
   <c sc="15" dc="15" oc="15" id="22746200">terminals</c>
   <c sc="11" dc="11" oc="11" id="22746290">terminology</c>
   <c sc="11" dc="11" oc="11" id="22746215">termination</c>
   <c sc="4" dc="4" oc="4" id="22746214">terminating</c>
   <c sc="2" dc="2" oc="2" id="22746152">termes</c>
   <c sc="2" dc="2" oc="2" id="22746218">terminator</c>
   <c sc="2" dc="2" oc="2" id="22746273">terminologies</c>
 </completions>
```

Figura 15: Ejemplo del uso de la variable **c**

En nuestro caso, utilizamos la URL para la consulta sobre publicaciones, la cual completamos con el parámetro que contiene el nombre del autor que queremos buscar (q) y el parámetro de formato, el cual recibe el valor 'JSON'.

Pues bien, como hemos explicado en el apartado donde se explicaban las tecnologías y herramientas utilizadas, para este caso hemos utilizado ciertas librerías que nos permiten abrir direcciones web y, por tanto, dicha información guardarla. La respuesta en formato JSON contiene una lista de diccionarios que es donde se sitúa la información que requerimos.

Dicha lista es recorrida y mediante las estructuras que nos facilita Python guardamos en diccionarios la información que deseamos guardar de esta consulta realizada. Se guardan el título del artículo, el lugar donde ha sido publicado dicho artículo y el tipo de artículo del que se trata.

Nos serviremos de estos datos recopilados para obtener información en las otras dos plataformas que hemos hablado en la introducción de este apartado. Para la búsqueda en Google Scholar nos serviremos del nombre del autor y de los diferentes títulos de los artículos. Por otro lado, los otros datos (el tipo de artículo y el lugar donde se publicaron los artículos) nos servirán para sacar de los *datasets* [4] donde se encuentran los indicadores científicos.

[4] La palabra *dataset* se trata de un término anglosajón empleado para referirse a los conjuntos de datos.

5.1.2. Extracción de información de Google Scholar

Como hemos expuesto antes, una de las razones por la que buscamos en información en Google Scholar es por el archivo encontrado en GitHub el cual ofrece la posibilidad de hacer consultas por comandos en la terminal del sistema.

Resumidamente, el archivo original (*scholar.py*) ofrecía distintas opciones para la búsqueda las cuales se pasaban como parámetros al método que realizaba la consulta a la plataforma y recibía la respuesta atendiendo a los valores escritos en la terminal. En nuestro caso realizamos ciertos cambios para que los valores no se pasaran por la terminal del sistema, sino que se pasaran atendiendo a lo que el usuario quisiera buscar.

En nuestra situación, fijamos que los valores que el usuario ha de elegir son básicamente el nombre del autor a buscar y, una vez mostradas los artículos pertenecientes al autor, deberá seleccionar el título del artículo para conocer los datos de dicho artículo. En nuestra solución, nos basamos en componer un método que recibiera estos dos valores, que más adelante nos servirán para configurar la consulta.

En un principio, creamos un diccionario donde se recogían los valores que determinaban la configuración de la búsqueda. Establecimos una configuración estándar, donde solo variaban los valores del nombre del autor y el valor '*phrase*', el cual, es utilizado para devolver los resultados que contengan exactamente dicho valor.

```
options =   {'cluster_id': None, 'author': None, 'allw': None, 'some': None, 'none': None, 'phrase': None,
            'title_only': False, 'pub': None, 'after': None, 'before': None, 'no_patents': True, 'no_citations': True,
            'citation': 'bt', 'count': 100000}
```

Figura 16: Diccionario de la configuración de la búsqueda en Google Scholar

Como se aprecia se tienen en cuenta todos los factores que se tenían en cuenta para la configuración del documento original. En la imagen se ve como tienen en valor *None* tanto el nombre del autor (clave *author*), como la clave *phrase*, sin embargo, tomarán los valores que recibe el método para establecer la configuración correctamente. Cabe destacar el parámetro *citation*, el cual indica el formato para citar los datos que devuelva la consulta. En este caso, escogimos la forma estándar que es *BibTeX*, en nuestro diccionario de configuración esta recogida con el valor *bt*. Otro aspecto por destacar en la figura mostrada superiormente es el valor de la clave *count* el cual recoge un valor de 100000. Como se explicó en puntos anteriores, el elemento de configuración *count* determina el número

máximo de resultados a mostrar, por tanto, elegimos un valor grande para así mostrar todos los resultados posibles en una misma consulta.

Una vez establecidas las opciones para la consulta se realiza una llamada a un método (*csv*) que consigue transformar los datos obtenidos por la respuesta que nos proporciona la API a una estructura CSV, utilizando como separador para diferenciar los datos el carácter '|'. De esta forma, sabiendo el orden de cómo ha sido estructurado los datos, podemos diferenciar a que referencia cada valor.

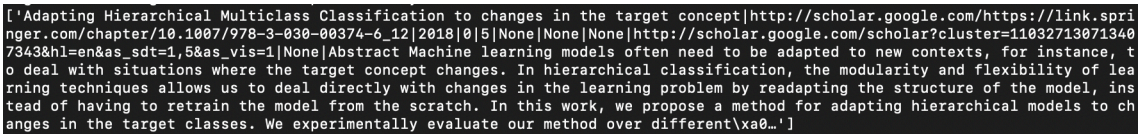


Figura 17: Resultado del método csv

En la figura 17 se muestra un claro ejemplo de los resultados que devuelve el método *csv*. Para este ejemplo se han utilizado como valores el nombre 'Cesar Ferri' (profesor en la UPV y tutor de este proyecto) y '*Adapting Hierarchical Multiclass Classification to changes in the target concept*' (publicación de Cesar Ferri).

El orden por el cual están ordenados los resultados que nos devuelve la consulta es el siguiente:

1- Título de la publicación
2- URL
3- Año de publicación
4- Número de citas que ha tenido del artículo
5- Número de versiones que tiene el documento
6- Identificador del documento dentro del servidor, conocido como *cluster id*
7- URL donde se encuentra el fichero PDF
8- URL de las versiones que tiene dicha publicación
9- URL de las citas que tiene el artículo
10- Resumen del artículo

Para tener en una estructura que nos permita obtener la información de una manera más sencilla, acudimos a las estructuras que nos ofrece Python y, al igual que en la búsqueda realizada en DBLP, los datos serán guardados en un diccionario. A la hora de realizar esto, nos ayudamos en un método, el cual creamos, que mediante la función *split* de

las librerías de Python nos devolviera una lista de los resultados quitando el separador utilizado por el método *csv*. Cada dato se guarda en una posición de la lista y en el orden en el que se recibe desde un principio. Finalmente, recorremos dicha lista y los datos son guardados en el diccionario mencionado y se devuelven como respuesta del método.

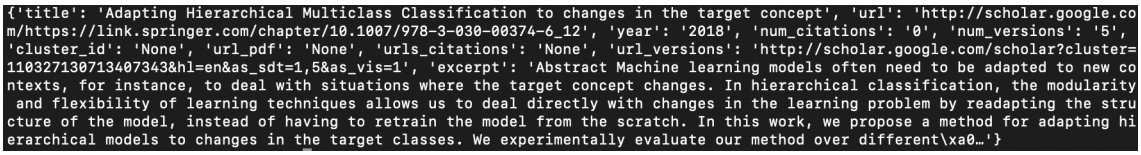


Figura 18: Resultados al guardar en el diccionario los datos.

Tanto en la figura 17 y en la figura 18 se pueden observar parámetros que tienen como valor *None*, este valor se da debido a que la consulta no obtiene valor para dicho parámetro, por tanto, le asigna ese resultado. Estos son la mayoría de los datos que serán visualizados en el entorno web.

5.1.3. Extracción de indicadores científicos

Terminando con la extracción de datos vamos a pasar a hablar de la extracción realizada sobre los *datasets* obtenidos de Scopus y GII-GRIN-SCIE (GGS) Conference Rating. Hablamos de *datasets* en plural debido a que tenemos un fichero diferente atendiendo al *venue* de la publicación, para este punto hemos tenido en cuenta únicamente cuando se trata de una conferencia o una revista. Para está función hemos creado un script que contenga los dos métodos que nos retornarán las puntuaciones que queremos obtener. Atendiendo al lugar donde fue publicada la publicación, una revista o una conferencia, los métodos recibirán el nombre de del sitio de publicación a buscar y devolverán las métricas. Estos datos de los que estamos hablando, y de los cuales nos servimos en este punto del proyecto, recordamos que fueron extraídos en la búsqueda en DBLP. A continuación, expondremos la extracción de los datos para la conferencia y, posteriormente, la extracción de los datos de las revistas.

Extracción de métricas para conferencias

El conjunto de datos del cual se extrae esta información proviene de la página web de *The GGS Conference Rating*, una herramienta con fin de evaluar los congresos de temática informática [22]. Dicha web ofrece un enlace de descarga directa para poder observar los datos y las calificaciones que ofrece.

Para tratar dicho archivo (**GII-GRIN-SCIE-Conference-Rating-30-mag-2018-11.54.45-Output.xlsx**) hemos realizado un método el cual tiene la función de recibir el nombre de la conferencia a buscar, recopilar las métricas y devolverlas. Primero, al tener el archivo en local, hemos tenido que localizar la ubicación donde se almacena éste. Una vez obtenida la dirección del directorio donde se encuentra, mediante la importación de las librerías que nos facilitan la lectura de datos en archivos compatibles con el nuestro, creamos el lector que nos facilitaría la búsqueda. Dicho lector, trata el archivo como lo que es, una matriz, y sabiendo exactamente las columnas correctas donde se encuentran los datos y recorriendo cada registro del fichero, obtuvimos las métricas que queríamos conseguir. De todas las columnas que tiene el archivo solo tenemos en cuenta tres de ellas: la columna 'GGS Class', la columna 'GGS Rating' y la columna 'Qualified Classes'.

	Title	Acronym	GGS Class	GGS Rating	Qualified Classes
0	3-D DIGITAL IMAGING AND MODELLING	3DIM	**Not Rated**	ated (discontinued	CORE:C
1	INTERNATIONAL CONFERENCE ON 3D IMAGING, MODELING, PROCESSING, VISUALIZATION &	3DIMPVT	**Work in Progress**	Work in Progress	MA:C
2	INTERNATIONAL SYMPOSIUM ON 3D DATA PROCESSING VISUALIZATION AND TRANSMISSION	3DPVT	**Work in Progress**	Work in Progress	CORE:C, MA:B

Figura 19: Estructura del archivo de las columnas utilizadas en el desarrollo

Como se aprecia en la figura 19 las columnas están dispuestas de esta forma, dando a ver que podemos realizar la búsqueda mediante el nombre de la conferencia (columna **Title**) o mediante el acrónimo de la conferencia (columna **Acronym**). Ahora bien, hay casos en la información recibida en la búsqueda de DBLP que no está contemplada en los conjuntos de datos, es decir, en el archivo no están todas las conferencias posibles que pueden estar en almacenadas en DBLP. Es por ello por lo que en caso de que la extracción de los datos deseados no recoja las puntuaciones deseadas, le hemos atribuido el valor 'Not Found' a dichos parámetros. Finalmente, una vez más, nos servimos de un diccionario para recoger los datos y devolverlos.

Extracción de métricas para revistas

Al igual que con las métricas de las conferencias, nos basaremos en un *dataset* (**JCR2018.xlsx**) obtenido directamente de Scopus, que ha sido almacenado en local, para conseguir las métricas. Pues bien, el funcionamiento del método que consigue estas métricas tiene ciertas similitudes con el método extractor de las métricas de las conferencias. Se obtiene la ubicación del directorio donde se almacena el archivo, se instancia el lector que nos facilitará la selección de las puntuaciones, se realiza la búsqueda atendiendo al nombre de la revista que recibe el método y se devuelven los resultados. Sin

embargo, para la realización de la búsqueda tuvimos que realizar un estudio sobre cuál era la mejor forma de conseguir los datos.

Sourcerecord id	Source Title (Medline-sourced journals are indicated in Green) Titles indicated in bold red do not meet the Scopus quality criteria anymore and therefore Scopus discontinued the forwar	2015 CiteScore	2015 SJR	2015 SNIP	2016 CiteScore	2016 SJR	2016 SNIP	2017 CiteScore	2017 SJR	2017 SNIP
18500162600	21st Century Music	ENG								
21100404576	2D Materials	5.89	4.602	1.009	4.26	2.314	0.915	6.05	2.813	1.072
21100447128	3 Biotech	ENG	0.145	0.119	2.15	0.462	1.199	2.23	0.511	1.033
21100779062	3D Printing and Additive Manufacturing	ENG	0.388	2.100	0.80	0.547	1.306	2.31	0.808	1.301

Figura 20: Estructura del archivo de las columnas utilizadas en el desarrollo

En la figura 20 se muestran las columnas utilizadas en el rastreo realizado por los registros del archivo. Como se puede ver, la búsqueda en el conjunto de datos se puede realizar por dos campos: un identificador que aportan a cada revista (**Soucerecord id**) o el nombre específico de la revista (**Source Title**). No obstante, hemos comentado que para la realización del rastreo hemos realizado un estudio con el fin de seleccionar la mejor forma. La causa de este estudio es debida a que los resultados devueltos por DBLP no son del todo convincentes. Cuando la publicación ha sido publicada en una revista, se devuelve el nombre de la revista con abreviaciones. Por ejemplo, para la revista 'Journal of Computational Science' se obtiene de resultado: 'J. Comput. Scien.'.

Es por ello por lo que al no poder realizar un rastreo fiable debido a que no teníamos el identificador de la revista ni el nombre completo de la revista decidimos recurrir al uso de utilizar a funciones que realizaran una comparación de la secuencia de los caracteres entre los registros del archivo y el nombre de la revista abreviado y devolviera el resultado más aproximado.

5.2. Caso de estudio: Selección de un predictor para el rastreo

Para comenzar a relatar el estudio que hemos llevado a cabo, introduciremos primeramente cuales han sido los predictores que hemos escogido. Como expusimos en el apartado de las tecnologías y herramientas empleadas, nos hemos servido de dos librerías. Por un lado, hemos utilizado el módulo *distance* con el cual hemos podido calcular la distancia de Levenshtein, el coeficiente de Sorensen-Dice y el índice de Jaccard. Por otro lado, la librería *difflib* la cual compara secuencias de caracteres.

Distancia de Levenshtein

La distancia de Levenshtein o distancia de edición entre dos cadenas de caracteres A_1 y A_2, consiste en determinar el conjunto mínimo de operaciones que son necesarias para llegar a transformas la cadena de texto A_1 en A_2 y viceversa. Las operaciones contempladas como operaciones de edición son: eliminar un carácter, insertar un carácter o sustituir un carácter [16].

```
Sea m la longitud de la palabra s1
Sea n la longitud de la palabra s2
Sea D la matriz de tamano [m,n]

# PASO 1 - Comprobacion
Si (m) == 0:
  Devolver n y salir

Si (n) == 0:
  Devolver m y salir


# PASO 2 - Inicializacion
Desde i = 1 hasta m hacer:
  D[i,0] = i

Desde j = 1 hasta n hacer:
  D[0,j] = j


# PASO 3 - Calculo de la matriz
Desde i = 1 hasta m hacer:
    Desde j = 1 hasta n hacer:
        D[i,j] = minimo( D[i-1,j] + 1,
                         D[i,j-1] + 1,
                         D[i-1,j-1] + 1 si s1[i] != s2[j]
                                    + 0 si s1[i] == s2[j]
                       )

# PASO 4 - Resultado
D[m,n] es la distancia de Levenshtein
```

Figura 21: Pseudocódigo de la distancia de Levenshtein [16]

Teniendo en cuenta la longitud de las cadenas de texto A_1 y A_2 como $|A_1|$ y $|A_2|$ respectivamente, la distancia de edición se representa por $dist_{lev}(|A_1|, |A_2|)$ y tiene como premisa que se cumple que $0 \leq distl_{ev}(A_1,A_2) \leq \max(|A_1|, |A_2|)$. Como se puede ver en la figura x, su cálculo recae sobre el uso de una matriz de dimensiones $[|A_1| + 1, |A_2| + 1]$.

En conclusión, este algoritmo nos devolverá el número de cambios mínimos producidos para llegar a transformar el nombre de la revista incompleto que obtuvimos en el nombre de la revista aparentemente correcto.

Coeficiente de Sorensen-Dice

El coeficiente o índice de Sorensen-Dice es utilizado para conseguir la similitud que hay entre dos elementos, en nuestro caso será dos cadenas de texto. Este coeficiente, a

diferencia del de Levenshtein, solamente tiene en cuenta la falta de datos o la presencia de datos en ambas muestras [18]. En nuestro caso, se basará en la presencia o ausencia de bigramas[5] en las cadenas de caracteres [6].

Siendo A_1 y A_2 dos muestras a comparar, cuyas longitudes o tamaño es $|A_1|$ y $|A_2|$ respectivamente, y sea B el número de datos en común entre las muestras A_1 y A_2, decimos que el coeficiente de similitud de Sorensen-Dice se aplica de la siguiente forma:

$$QS = \frac{2B}{A1 + A2} = \frac{2\,|A1 \cap A2|}{|A1| + |A2|}$$

Como muestra la fórmula, el coeficiente de Sorensen-Dice puntúa de manera doble a aquellos elementos, o bigramas en nuestro caso, que tienen presencia. Este coeficiente estadístico alcanza valores entre [0, 1] de rango.

Índice de Jaccard

El índice de Jaccard mide la similitud entre dos conjuntos [18]. Al igual, que en el coeficiente de Sorensen-Dice, únicamente tiene en cuenta la presencia o la ausencia de elementos entre muestras. Teniendo en cuenta los mismos valores:

- A_1, representa un conjunto a comparar, una cadena de texto para nuestro estudio. Su tamaño o longitud es considerado como $|A_1|$.
- A_2, representa el otro conjunto a comparar, la otra cadena de texto a estudiar. Su tamaño o longitud es $|A_2|$.
- B, representa el número de elementos coincidentes en ambas muestras. Como ya hemos explicado antes, serán los bigramas coincidentes en ambas cadenas de texto.

El cálculo de este coeficiente se basa en la siguiente formula:

$$J(A1, A2) = \frac{B}{A1 + A2} = \frac{|A1 \cap A2|}{|A1| + |A2|}$$

[5] Se conoce como n-grama a la subsecuencia de n elementos en una secuencia. De este modo hablamos de bigrama cuando la subsecuencia de la que hablamos está formada por dos caracteres.

A diferencia del índice de Sorensen-Dice, el índice de Jaccard puntúa por igual a los términos presentes o ausentes en las muestras. El resultado de esta métrica estadística se es representado en un rango de [0, 1] [18].

Difflib

La librería de Python Difflib facilita una serie de clases y métodos cuyo fin es la comparación de secuencias. Concretamente, de entre las posibilidades que ofrece la librería, nos basamos en la clase *SequenceMatcher*.

El algoritmo comparador de esta clase consiste en, primeramente, identificar la cadena de caracteres coincidente más larga. Una vez identificada, aplicar recursivamente tanto a izquierda como a derecha de la subsecuencia el mismo procedimiento [17].

La clase contiene un método (***ratio()***) que aplicando el algoritmo explicado devuelve una ponderación entre [0, 1] de la similitud de ambas secuencias. Cuando el resultado de este método es igual o supera una ponderación de 0,6 se puede decir que las secuencias tienen alto grado de similitud. Este ha sido el método elegido para ponderar la similitud. La fórmula para extraer dicha métrica es muy parecida a las anteriores vistas. Siendo A el número total de caracteres de ambas secuencias y siendo B el número de coincidencias encontradas, la expresión es [17]:

$$ratio = \frac{2B}{A}$$

Presentación de muestras a estudiar

Todo estudio empírico necesita muestras o ejemplos a estudiar para extraer resultados concluyentes. Basándonos en el *dataset* que recopila las revistas donde que van a tener que ser comparadas, escogimos 10 revistas al azar, cuyo nombre fue abreviado también por nosotros. A continuación, se muestra una tabla con el nombre de la revista al completo (extraído exactamente igual que en el *dataset*) y en la parte derecha su abreviatura:

Revista	Abreviatura
Journal of Computational Science	J. Comput. Scien.
Data Mining and Knowledge Discovery	Data Min. Knowl. Discov.
Applied Intelligence	Appl. Intell.

Journal of Machine Learning Research	J. Mach. Learn. Res.
Copenaghen Journal of Asian Studies	Copen. Jo. As. Stud.
International Journal of Computing	Int. J. Comp.
Foundation and Trends in Databases	Found. Tren. Datab.
Handbook of Computational Economics	Hand. Comp. Econ.
Mathematicals Models and Computer Simulations	Math. Mod. Comp. Sim.
Quality Innovation Prosperity	Qual. Innov. Pros.

Desarrollo y factores del estudio

Para desarrollar el caso de estudio creamos un script que almacena los distintos métodos que nos devolverán las puntuaciones que deseamos obtener. Cada método corresponde a cada uno de los algoritmos explicados anteriormente y reciben como parámetro el nombre de la revista abreviado a buscar. Los métodos contienen una estructura idéntica:

1- Acceso al conjunto de datos donde leerán
2- Bucle que recorre los registros del archivo comparando y obteniendo una puntuación de la similitud. En este punto es donde se calcula el tiempo total de la ejecución del algoritmo.
3- Ordenación de los resultados para obtener las mejores puntuaciones.
4- Selección de los 10 mejores para puntuar el acierto del predictor.
5- Respuesta de esas 10 mejores puntuaciones.

Pues bien, los factores determinantes para la elección del predictor que hemos seleccionado son los siguientes:

- Tiempo de ejecución del algoritmo comparador.
- Veces que el resultado devuelto contenía el nombre de la revista correcta en el primer puesto.
- Veces que el resultado devuelto contenía el nombre de la revista correcta dentro de los primeros 5 puestos.
- Veces que el resultado devuelto contenía el nombre de la revista correcta dentro de los primeros 10 puestos.

Resultados

Los resultados han sido recopilados en una hoja de cálculo, donde cada algoritmo tiene una tabla con sus puntuaciones que más adelante nos han servido para sacar ciertas gráficas que nos pudieran facilitar una conclusión de forma más visible. Cada tabla recoge los siguientes parámetros:

- Nombre de la revista original
- Nombre de la revista abreviada
- Puntuación de Top 1: pertenece al Top 1 (1) y en caso negativo (0)
- Ratio del algoritmo de Top 1
- Puntuación de Top 5: pertenece al Top 5 (1) y en caso negativo (0)
- Ratio del algoritmo de Top 5
- Puntuación de Top 10: pertenece al Top 10 (1) y en caso negativo (0)
- Ratio del algoritmo de Top 10
- Tiempo de ejecución

Cabe destacar un apunte sobre las puntuaciones, en caso de que el resultado devuelva como resultado una revista que esté en el Top 1, a su vez, ésta misma será puntuada también como perteneciente al Top 5 y al Top 10. Se produce el mismo caso cuando el resultado devuelve una revista perteneciente al Top 5 pero no al Top 1, dicha revista también será puntuada positivamente en el Top 10.

Todos los resultados referentes a las puntuaciones de los algoritmos están detallados en sus tablas (Tabla 2, Tabla 3, Tabla 4 y Tabla 5) en el Anexo.

Discusión

Gracias a los resultados hemos realizado una serie de gráficas que hacen que los resultados sean más visibles para determinar una conclusión. Estas gráficas están basadas en los factores que en un principio determinamos. Las gráficas se muestran a continuación:

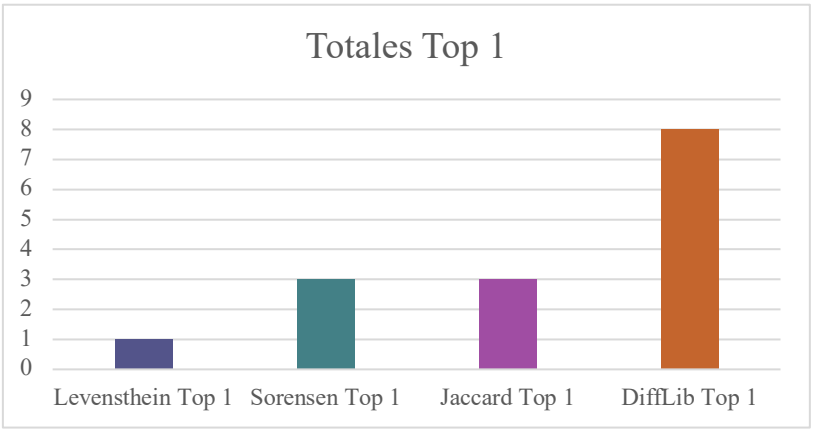


Figura 22: Gráfica que representa el número de Top 1 realizados por algoritmo

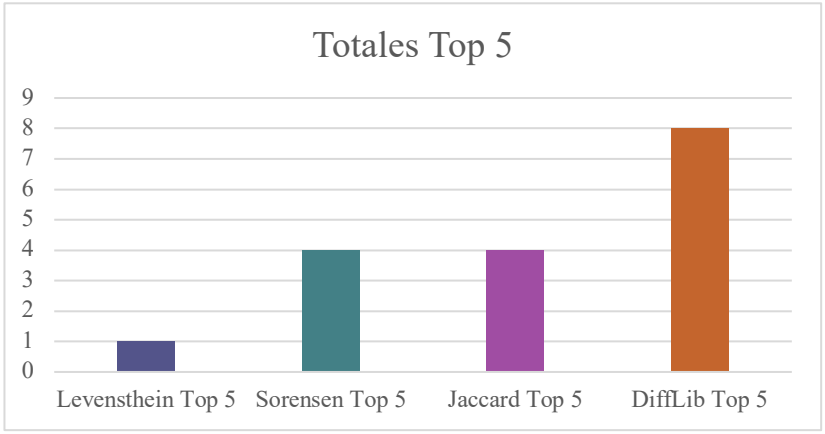


Figura 23: Grafica que representa el número de Top 5 realizados por algoritmo

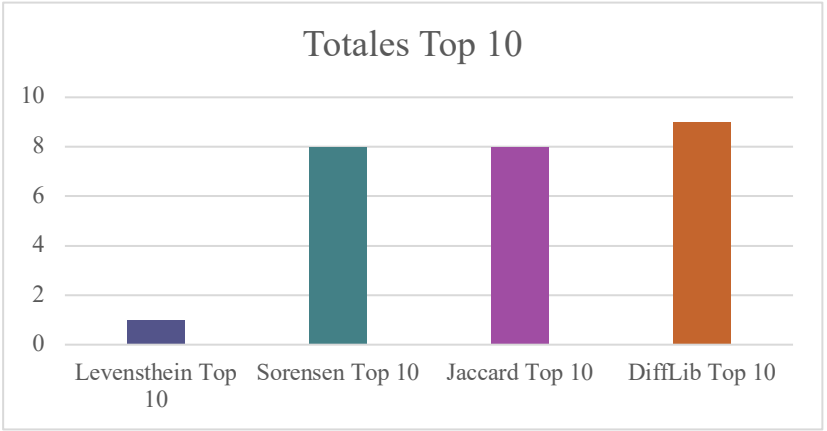


Figura 24: Gráfica que representa el número de Top 10 realizados por algoritmo

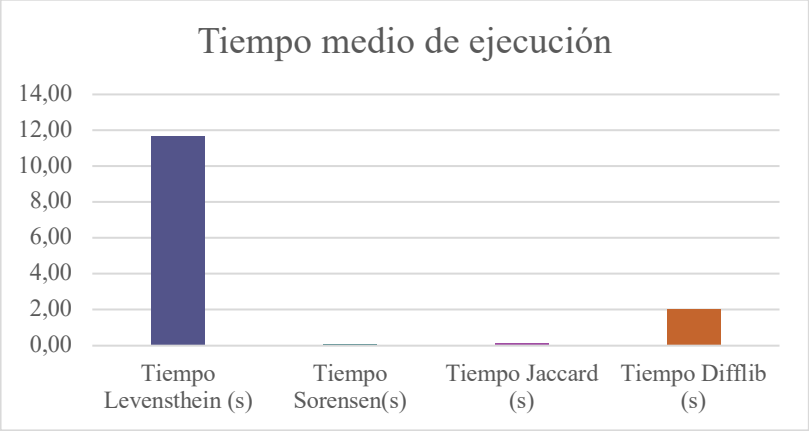


Figura 25: Gráfica que representa el tiempo medio de ejecución (s) por algoritmo

A la hora de valorar los puntos obtenidos por cada algoritmo en los distintos Top que hemos extraído hemos realizado una regla de valor sobre ellos:

1 punto de Top 1 > 1 punto de Top 5 > 1 punto de Top 10

Por tanto, podemos decir que es preferible obtener puntuación en el Top 1, debido a que la respuesta es la correcta. Pues bien, observando los resultados, salta a la vista que el ganador en el ámbito de acierto es la librería *Difflib*, ya que, obtiene un 80% de acierto al comparar secuencias. Además, cabe destacar que con clara diferencia con los otros algoritmos.

Por otro lado, en la gráfica del tiempo medio de ejecución se puede apreciar una gran diferencia entre algoritmos. Observamos que el algoritmo de la distancia de Levenshtein alcanza un tiempo medio muy elevado en comparación al de sus rivales (casi 12 segundos). También observamos los tiempos más bajos en los algoritmos de Sorensen-Dice y Jaccard, ésta podría ser la mejor opción. No obstante, la relación de acierto con el tiempo medio no es igual de buena que la que tiene el algoritmo utilizado por la librería *Difflib*.

En definitiva, la elección concluyente que hemos realizado para ser el predictor del nombre de la revista es la librería *Difflib* con su clase *SequenceMatcher*. Esta opción ha demostrado un alto ratio de acierto en las pruebas realizadas (80%) además de mostrar un tiempo medio bastante aceptable (2 segundos).

5.3. Creación del entorno web e implementación de la funcionalidad

La creación el entorno web la hemos realizado desde cero, creando la plantilla de lo que sería la apariencia de esta aplicación. Cabe destacar que la ejecución en el entorno web se reproducirá en local.

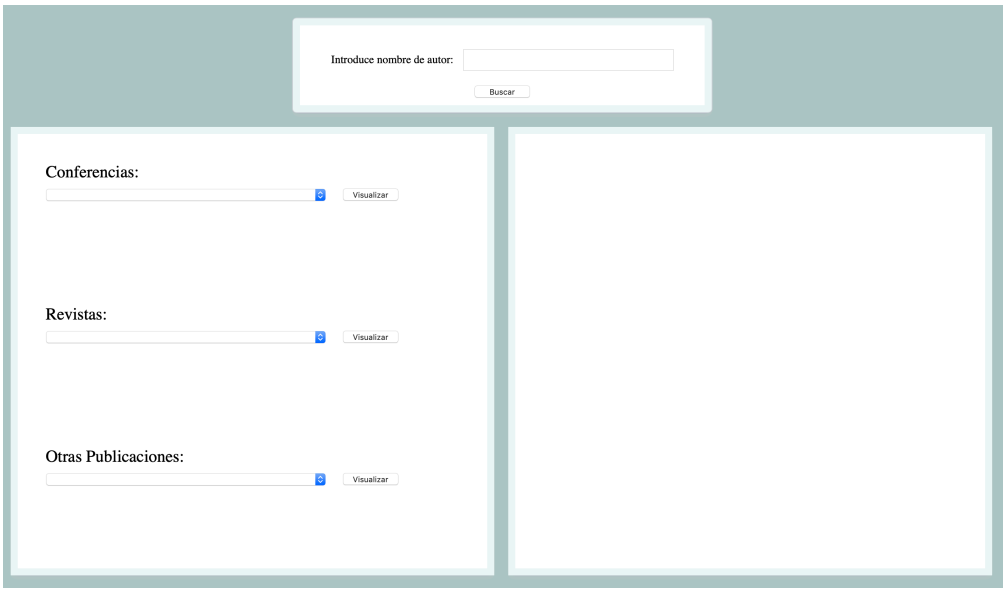


Figura 26: Apariencia del entorno web

Como se aprecia en la figura 26, en la interfaz se distinguen tres partes. Una parte donde se introduce el nombre del autor que se quiere buscar, una parte situada en la zona izquierda de la pantalla donde se mostrarán los resultados y, a la derecha de la pantalla, una parte dedicada a mostrar los datos de la publicación seleccionada.

Ahora bien, para añadir la funcionalidad a nuestra página web hemos recurrido a Flask. Flask es pequeño *framework* programado en Python, el cual nos permite de una manera sencilla conectar la interfaz del entorno web con las funciones que hemos creado. Para manejar este *framework* hemos creado un script (**render.py**), el cual contiene las instrucciones a realizar atendiendo a la solicitud que envía el usuario. Además, en este caso al ser una aplicación que en estos instantes funciona en local, este archivo configura el puerto por donde se producirá la comunicación (puerto 5000).

```
@app.route('/')
def index():
    return render_template('intro.html')
```

*Figura 27: Método que inicia la interfaz (archivo **intro.html**)*

El método *render_template()*,tiene como funcionalidad cargar de primera la plantilla recibida como parámetro. En nuestro caso, como se observa en la figura x, nuestra plantilla es *intro.html*. Los demás métodos que contiene este archivo *render.py* son los siguientes:

- **search()**: recupera las publicaciones del autor a buscar y las transmite a la interfaz.
- **viewConf()**: extrae los datos del artículo seleccionado de la caja de selección de la parte de conferencias.
- **viewJournal()**: extrae los datos del artículo seleccionado de la caja de selección de la parte de revistas.
- **viewOthers()**: extrae los datos del artículo seleccionado de la caja de selección de la parte de otros artículos, los cuales, no tendrán indicador científico y no hará búsqueda en ninguno de los dos conjuntos de datos.

Por otro lado, tenemos las funciones en la parte del archivo HTML, las cuales están asociadas a cada botón que hay en la interfaz. Para estas funciones nos hemos ayudado de una librería de JavaScript llamada jQuery, la cual es empleada especialmente para la interacción en páginas web. Las funciones de las que estamos hablando son:

- Al pulsar el **botón de 'Buscar'**: recoge el nombre del autor y lo pasa como parámetro para la extracción de las publicaciones de ese autor. Una vez recibida la respuesta, atendiendo al lugar donde ha sido publicado ese artículo, lo carga en las distintas listas de selección que hay presentes.
- Al pulsar el **botón de visualizar conferencias**: recoge el nombre del autor y el nombre del artículo seleccionado en la lista de selección y los pasa como parámetros a la función **viewConf()** del archivo *render.py*. Una vez recibe la respuesta, muestra los resultados en la parte de la derecha de la interfaz.
- Al pulsar el **botón de visualizar revistas**: recoge el nombre del autor y el nombre del artículo seleccionado y los transmite a la función **viewJournal()**.

Al recibir una respuesta, plasma esos resultados en la parte donde se muestran los resultados.

- Al pulsar el **botón de visualizar otros**: recoge el valor del autor a buscar y el artículo seleccionado y envía estos parámetros a la función **viewOthers()**, la cual, le envía los datos necesarios y ésta los muestra.

Para dejar más claro toda la transmisión de datos que se produce en la aplicación, vamos a mostrar un esquema representativo de cómo es el flujo de datos de ésta al buscar los datos de un autor e intentar buscar los datos de un documento.

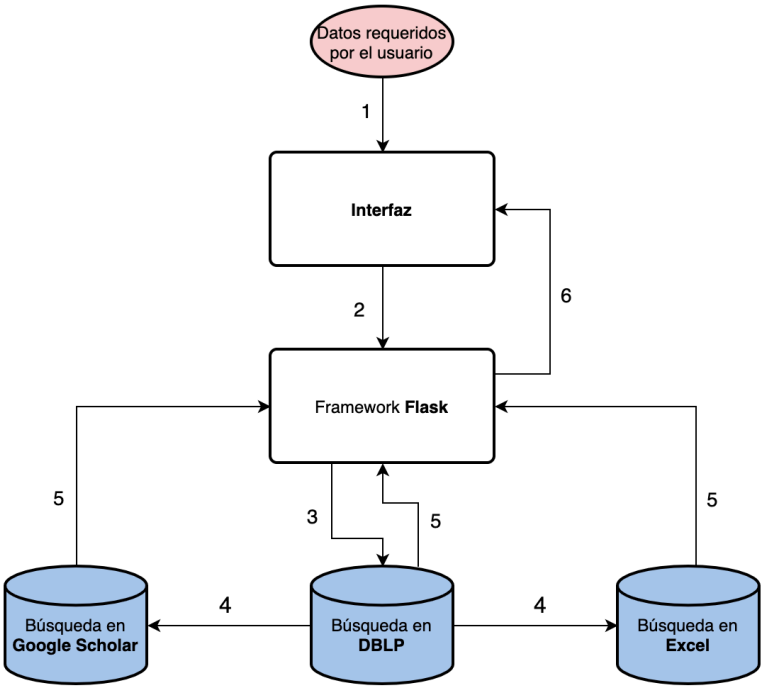


Figura 28: Esquema de los pasos que sigue la aplicación

6. Implantación y resultados

La aplicación desarrollada como hemos detallado anteriormente ha sido creada para que en un principio funcione a nivel local, es decir, no está accesible vía internet de momento. En cuanto al código del proyecto, han sido subidos a un repositorio en la plataforma GitHub (https://github.com/ivcarma/TFG) los scripts relacionados con la extracción de datos. Hacer visible a todo el mundo la aplicación puede ser un punto por tratar en caso de planificar un futuro proyecto.

En cuanto a los resultados del proyecto, vamos a explicar cuál es el funcionamiento de la aplicación de una forma más visual. Para ello hemos de elegir un autor para buscar, en nuestro caso, buscaremos las publicaciones hechas por Cesar Ferri, uno de los tutores de este proyecto.

Primeramente, daremos paso a mostrar la interfaz de la aplicación. Se basa de una pantalla en la cual se destacan tres zonas características: la zona donde se escribe el nombre del autor a buscar, la zona donde aparecerán las publicaciones clasificadas atendiendo al lugar donde han sido publicadas y la zona de los resultados.

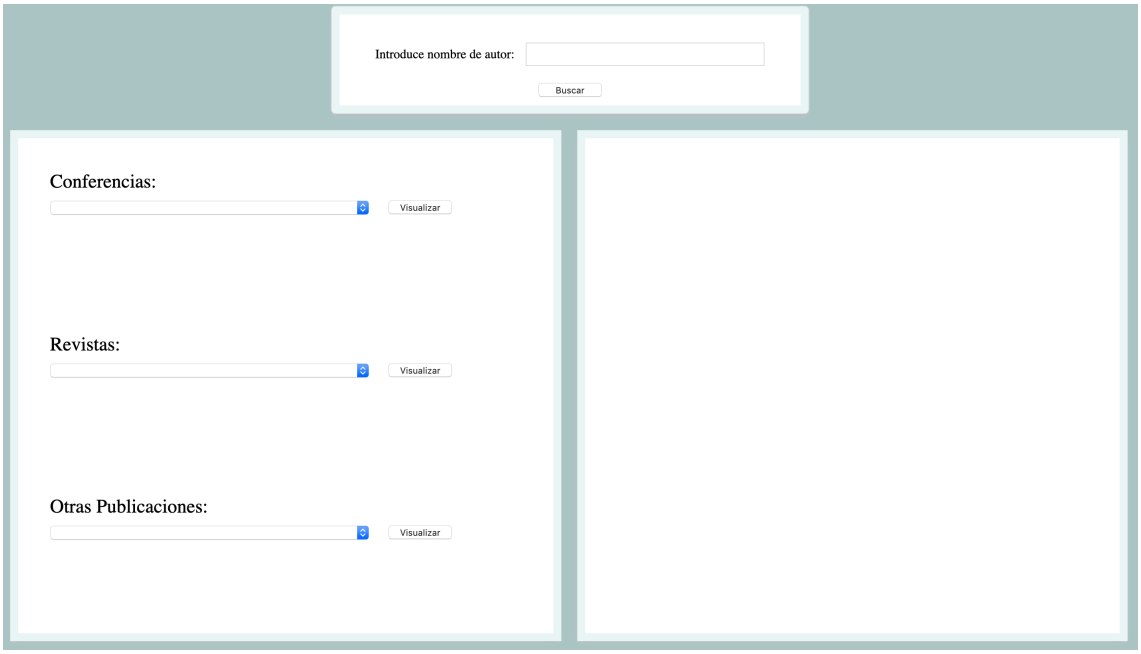


Figura 29: Pantalla de inicio de la aplicación.

Seguidamente, empezaremos la búsqueda de las publicaciones, para ello introduciremos el nombre del autor “Cesar Ferri” y pulsaremos el botón de buscar.

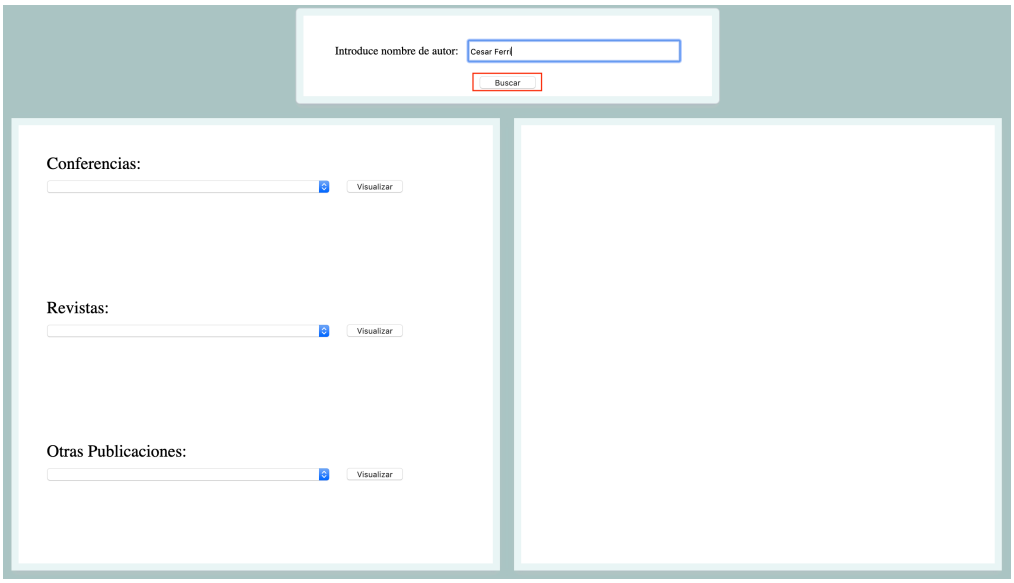


Figura 30: Pantalla al introducir nombre del autor

Por consiguiente, la aplicación nos mostrará las publicaciones de dicho autor, clasificando dichas publicaciones dentro del grupo que les pertenece. Cabe destacar que el grupo catalogado como ‘Otras Publicaciones’ va dedicado a todas aquellas publicaciones que no han sido expuestas ni en conferencias ni han formado parte de revistas.

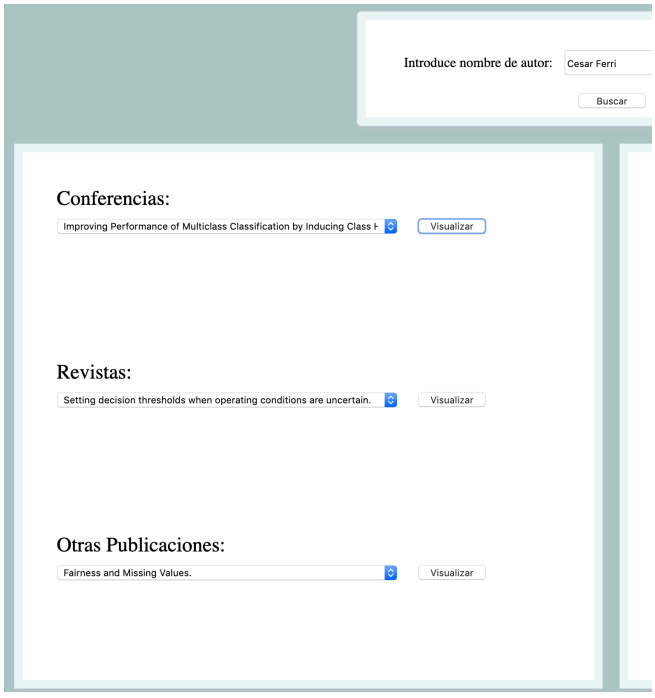
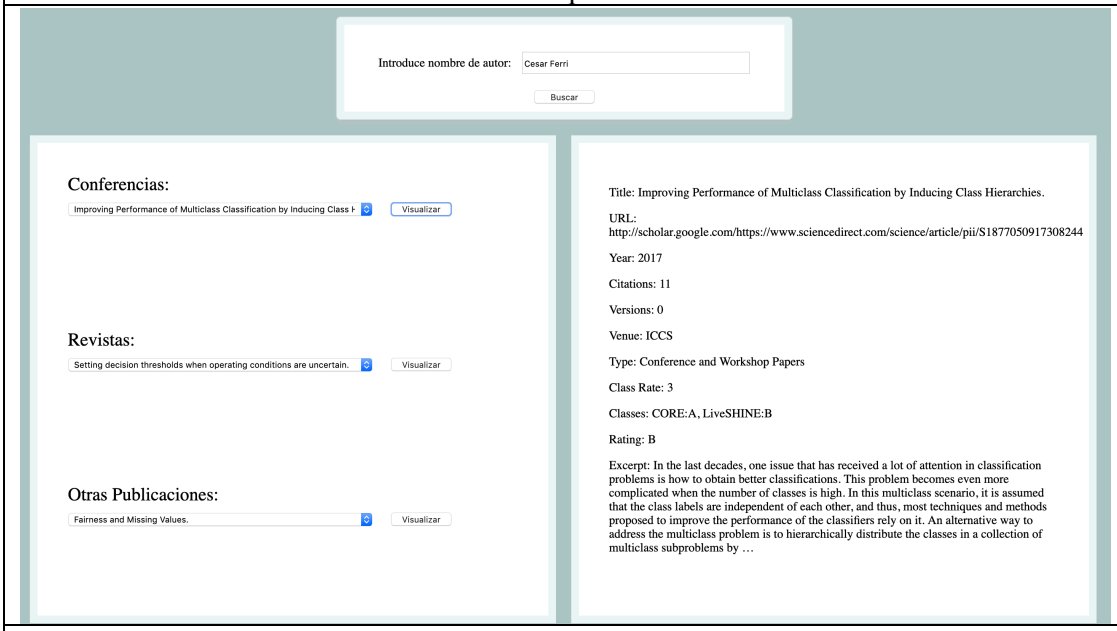


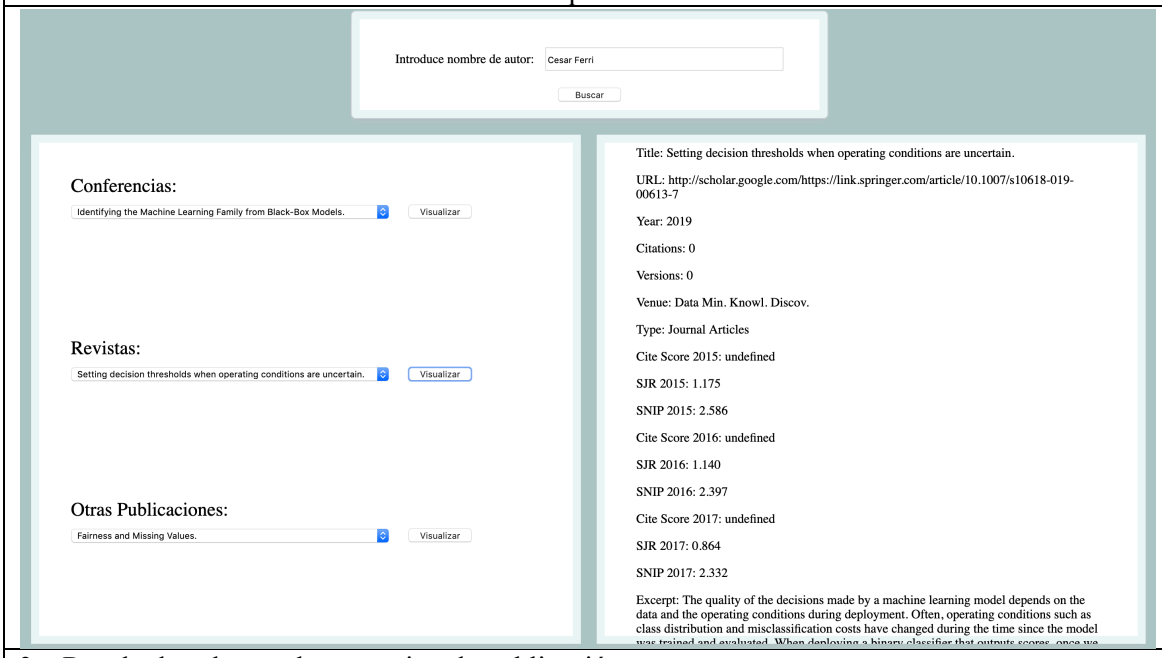
Figura 31: Clasificación de las publicaciones atendiendo a su venue.

Llegados a este punto, el usuario podrá decidir qué publicación consultar. Para visualizar los resultados bastará con seleccionar el título del documento deseado y pulsar el botón de 'Visualizar' que corresponde. Cada grupo de títulos tiene asociado un botón 'Visualizar' posicionado a la derecha de cada lista. Los resultados pueden verse en la tabla mostrada inferiormente:

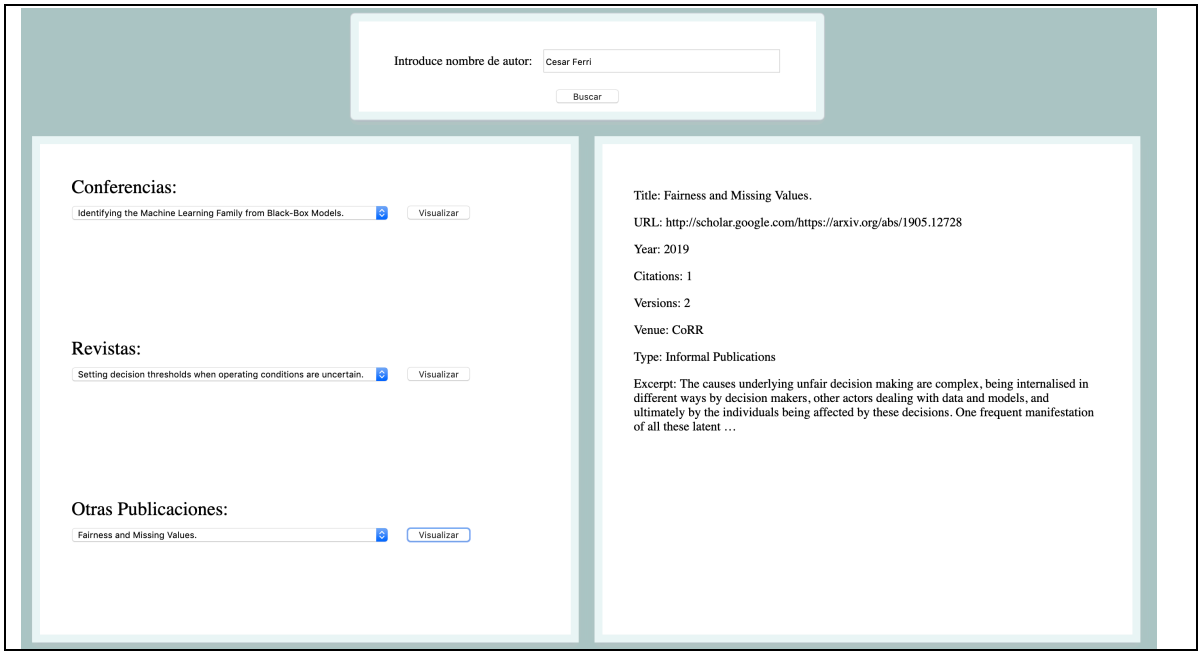
1- Resultados al consultar un documento publicado en una conferencia



2- Resultados al consultar un documento publicado en una revista



3 – Resultados al consultar otro tipo de publicación



Como se puede ver en la tabla situada superiormente, cada tipo de consulta devuelve un tipo de resultado, esto se debe a que cada tipo de publicación obtiene unos indicadores blibliométricos distintos.

7. Conclusiones y trabajo futuro

Como dijimos al principio de esta memoria, el objetivo principal ha sido realizar la aplicación web que hemos detallado a lo largo de este proyecto. Pues bien, se puede decir que el objetivo propuesto ha sido conseguido, pero igual no de la forma que esperábamos.

A lo largo del desarrollo nos hemos encontrado con varios obstáculos que no tuvimos en cuenta. Los obstáculos de los que hemos hablado los hemos encontrado a la hora de extraer la información de las bases de datos de Scopus y GII-GRIN-SCIE (GGS) Conference Rating.

Por un lado, a la hora de extraer las métricas relativas a las conferencias nos dimos cuenta de que había casos en los que no encontraba resultado. La información que nos aporta el nombre de la conferencia es extraída de DBLP, donde engloba en la misma categoría a los documentos publicados en conferencias o en seminarios o talleres. Es por ello por lo que se puede dar el caso que se devuelva un nombre de algún seminario o taller no indexado o que no aparezca en rankings internacionales y este no se encuentre en los conjuntos donde trabaja la aplicación para extraer los indicadores. La solución que dimos es denotar con el valor '*None*' a aquellos resultados que no se encuentren.

Por último, tuvimos también un obstáculo al buscar las métricas de las revistas. Nos encontramos con un problema parecido como con las conferencias, pero en este caso debido al formato en la que se nos devolvía el nombre de la revista. Al realizar la extracción en DBLP el nombre de la revista se nos devolvía abreviado, por tanto, al hacer la búsqueda no se encontraban resultados. Para hallar una solución, realizamos un caso empírico de estudio en el cual, de entre las opciones que teníamos, escogimos un método predictor que devolviera la información que según su algoritmo correspondía con la del nombre abreviado.

Los problemas causados han sido en parte debidos a problemas con el formato de los datos comparados. No obstante, esto abre la puerta a futuras mejoras para un futuro proyecto donde se quiera profesionalizar esta aplicación. Por ello algunas propuestas para un futuro proyecto podrían ser:

- Mejorar la interfaz de tal forma que sea más atractiva de cara al usuario.

- Obtener solución para obtener con certeza las métricas de las conferencias y de las revistas.
- Establecer alguna fuente de información que nos aporte la información que nos aporta DBLP, pero de todos los autores posibles, no solo autores de carácter informático.

En definitiva, creo que el proyecto realizado puede tener diversas utilidades. Por una parte, se le puede asociar una labor evaluativa, por ejemplo, a la hora de evaluar investigadores, esto podría servir en el caso de que se quisiera definir ciertas plazas de un tribunal, lo que ayudaría a catalogar que investigadores serían los idóneos. Por otro lado, también se le puede otorgar una función informativa, ya que, muestra al usuario datos bibliográficos que pueden ofrecerle algún tipo de ayuda o conocimiento.

8. Referencias

[1] - *API Google Scholar Github*. Obtenido de https://github.com/ckreibich/scholar.py

[2] - Ardanuy, J. (2012). *Breve introducción a la bibliometría.*

[3] - Challenger Perez, I., Díaz Ricardo, Y., & Becerra García, R. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 1-13.

[4] - *CiteSeer*. Obtenido de http://csxstatic.ist.psu.edu/home

[5] - Codina, L. (2005). Scopus: el mayor navegador científico de la web. *El profesional de la información*.

[6] - *Coeficiente de Dice*. Obtenido de https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Dice%27s_coefficient

[7] - *CORE*. Obtenido de http://www.core.edu.au/conference-portal

[8] - *DBLP F.A.Q.* Obtenido de DBLP : https://dblp.uni-trier.de/faq/

[9] - *DBLP Statistics*. Obtenido de https://dblp.uni-trier.de/statistics/

[10] - Eloisa Vargiu, M. U. (2013). Exploiting web scraping in a collaborative filteringbased approach to web advertising. *Journal of Artificial Intelligence Research*.

[11] - *Flask*. Obtenido de https://palletsprojects.com/p/flask/

[12] - *Gruppo di Informatica (GRIN)*. Obtenido de http://www.grin-informatica.it

[13] - *Gruppo di Ingegneria Informatica (GII)*. Obtenido de http://www.gii.it/

[14] - Guerrero Bote , V., & Moya Anegón, F. (2012). A further step forward in measuring journals' scientific prestige: The SJR2 indicator. *Journal of Informetrics*.

[15] - Martín Martín, A., Orduna Malea, E., Anne Wil, H., & López Cózar, E. (2017). Can we use Google Scholar to identify the highly-cited documents? *Journal of Informetrics*.

[16] - Pérez Melián, J. (2016). *Análisis de frecuencia de hashtags en Twitter.*

[17] - Python. *The Python Standard Library*. Obtenido de https://docs.python.org/2/library/

[18] - Rodríguez Salazar, M., Álvarez Hernández, S., & Bravo Núñez, E. (2000). *Coeficientes de asociación.*

[19] - *Scopus F.A.Q.* Obtenido de https://service.elsevier.com/app/answers/list/

[20] - Sirisuriya, S. d. (2015). A Comparative Study on Web Scraping. *International Research Conference, KDU.*

[21] - *Sociedad Científica Informática de España (SCIE)*. Obtenido de http://www.scie.es/

[22] - *The GII-GRIN-SCIE (GGS) Conference Rating*. Obtenido de http://gii-grin-scie-rating.scie.es/conferenceRating.jsf

[23] - Torres Salinas, D., Ruiz Pérez, R., & López Cózar, E. (2009). Google Scholar como herramienta para la evaluación científica. *El profesional de la información*.

[24] - Velho, L. (1993). *Indicadores Científicos: Aspectos Téoricos y Metodológicos.*

[25] - Villota García, S., Zamora López, G., & Llanga Vargas, E. (2019). Uso de Internet como base para el aprendizaje. *Atlante: Cuadernos de Educación y Desarrollo*.

Anexo

Original Classes	Final Class	Original Classes	Final Class	Original Classes	Final Class	Original Classes	Final Class
A++, A++, A++	A++	A++, A+, B-	A	A+, A-, C	B	A+, C, C	B-
A++, A++	A	A++, A, B	A	A+, B, B-	B	A+, C	B-
A++, A++, A+	A++	A++, A-, A-	A	A, A, C	B	A-, B-, C	B-
A++, A++, A	A+	A++, A-	A	A, A-, B-	B	B, B, C	B-
A++, A+, A+	A+	A, A, A-	A	A, B, B	B	B, B-, B-	B-
A++, A+	A	A++, A+, C	A-	A, B	B	B, B-	B-
A++, A++, A-	A+	A++, A, B-	A-	A++, B, C	B	A, C, C	B-
A++, A+, A	A+	A++, A-, B	A-	A++, B-, B-	B	A, C	B-
A++, B, B	A-	A+, A+, B-	A-	A++, B-	B	B, B-, C	B-
A++, C, C	B	A+, A, B	A-	A, A-, C	B	B-, B-, B-	B-
A++, B, B-	A-	A+, A-, A-	A-	A, B, B-	B	B-, B-	B-
A++, A-, A-	A	A+, A-	A-	A-, A-, B-	B	A-, C, C	B-
A++, A++, B	A	A+, A+, C	A-	A-, B, B	B	A-, C	B-
A++, A+, A-	A	A+, A, B-	A-	A-, B	B	B-, B-, C	B-
A++, A, A	A	A+, A-, B	A-	A++, B-, C	B	B, C, C	Work in Progress
A++, A	A	A, A, B	A-	A+, B, C	B	B, C	Work in Progress
A++, A++, B-	A	A, A-, A-	A-	A+, B-, B-	B	B-, C, C	Work in Progress
A++, A+, B	A	A, A-	A-	A+, B-	B	B-, C	Work in Progress
A++, A, A-	A	A++, A, C	A-	A-, A-, C	B	B, NC	Work in Progress
A+, B, C	B	A++, A-, B-	A-	A-, B, B-	B	C, C, C	Work in Progress
A+, B, B	A-	A++, B, B	A-	B, B, B	B	C, C, NC	Work in Progress
A+, B	A-	A++, B	A-	B, B	B	B, C, NC	Work in Progress
A+, A+, A+	A+	A, A, B-	A-	A+, B-, C	B	C, NC	Work in Progress
A+, A+	A	A, A-, B	A-	A, B, C	B	C, C	Work in Progress
A+, A+, A	A+	A-, A-, A-	A-	A, B-, B-	B	A	Work in Progress
A+, A+, A-	A	A-, A-	A-	A, B-	B	A+	Work in Progress
A+, A, A	A	A++, A-, C	A-	A++, C, C	B	A++	Work in Progress
A+, A	A	A++, B, B-	A-	A++, C	B	A-	Work in Progress
A++, A++, C	A	A+, A, C	A-	B, B, B-	B	B	Work in Progress
A+, A+, B	A	A+, A-, B-	A-	A, B-, C	B-	B-	Work in Progress
A+, A, A-	A	A+, B, B	A-	A-, B, C	B-	C	Work in Progress
A, A, A	A	A+, B	A-	A-, B-, B-	B-	NC	Work in Progress
A, A	A	A-, A-, B	A-	A-, B-	B-		

Tabla 1: Combinaciones para las calificaciones del algoritmo de GGS [12]

Revista	Abreviatura	Levensthein Top 1	Ratio Levensthein Top 1	Levensthein Top 5	Ratio Levensthein Top 5	Levensthein Top 10	Ratio Levensthein Top 10	Tiempo ejecución (s)
Journal of Computational Science	J. Comput. Scien.	0		0		0		10,64
Data Mining and Knowledge Discovery	Data Min. Knowl. Discov.	1	14	1	14	1	14	14,83
Applied Intelligence	Appl. Intell.	0		0		0		8,33
Journal of Machine Learning Research	J. Mach. Learn. Res.	0		0		0		12,59
Copenaghen Journal of Asian Studies	Copen. Jo. As. Stud.	0		0		0		12,52
International Journal of Computing	Int. J. Comp.	0		0		0		9,5
Foundation and Trends in Databases	Found. Tren. Datab.	0		0		0		11,97
Handbook of Computational Economics	Hand. Comp. Econ.	0		0		0		10,7
Mathematicals Models and Computer Simulations	Math. Mod. Comp. Sim.	0		0		0		12,87
Quality Innovation Prosperity	Qual. Innov. Pros.	0		0		0		12,72

Tabla 2: Resultados del algoritmo de la distancia de Levenshtein

Revista	Abreviatura	Sorensen Top 1	Ratio Sorensen Top 1	Sorensen Top 5	Ratio Sorensen Top 5	Sorensen Top 10	Ratio Sorensen Top 10	Tiempo ejecución (s)
Journal of Computational Science	J. Comput. Scien.	0		0		1	0.161	0,11
Data Mining and Knowledge Discovery	Data Min. Knowl. Discov.	1	0.176	1	0.176	1	0.176	0,11
Applied Intelligence	Appl. Intell.	0		1	0.238	1	0.238	0,09
Journal of Machine Learning Research	J. Mach. Learn. Res.	0		0		1	0.225	0,11
Copenaghen Journal of Asian Studies	Copen. Jo. As. Stud.	0		0		1	0.235	0,1
International Journal of Computing	Int. J. Comp.	0		0		1	0.241	0,1
Foundation and Trends in Databases	Found. Tren. Datab.	1	0.103	1	0.103	1	0.103	0,1
Handbook of Computational Economics	Hand. Comp. Econ.	0		0		0		0,1
Mathematicals Models and Computer Simulations	Math. Mod. Comp. Sim.	0		0		0		0,11
Quality Innovation Prosperity	Qual. Innov. Pros.	1	0.199	1	0.199	1	0.199	0,11

Tabla 3: Resultados del algoritmo del coeficiente de Sorensen-Dice

Revista	Abreviatura	Jaccard Top 1	Ratio Jaccard Top 1	Jaccard Top 5	Ratio Jaccard Top 5	Jaccard Top 10	Ratio jaccard Top 10	Tiempo ejecución (s)
Journal of Computational Science	J. Comput. Scien.	0		0		1	0.278	0,14
Data Mining and Knowledge Discovery	Data Min. Knowl. Discov.	1	0.3	1	0.3	1	0.3	0,15
Applied Intelligence	Appl. Intell.	0		1	0.385	1	0.385	0,13
Journal of Machine Learning Research	J. Mach. Learn. Res.	0		0		1	0.368	0,13
Copenaghen Journal of Asian Studies	Copen. Jo. As. Stud.	0		0		1	0.381	0,15
International Journal of Computing	Int. J. Comp.	0		0		1	0.389	0,13
Foundation and Trends in Databases	Found. Tren. Datab.	1	0.188	1	0.188	1	0.188	0,14
Handbook of Computational Economics	Hand. Comp. Econ.	0		0		0		0,19
Mathematicals Models and Computer Simulations	Math. Mod. Comp. Sim.	0		0		0		0,13
Quality Innovation Prosperity	Qual. Innov. Pros.	1	0.333	1	0.333	1	0.333	0,13

Tabla 4: resultados del algoritmo del índice de Jaccard

Revista	Abreviatura	DiffLib Top 1	Ratio DiffLib Top 1	DiffLib Top 5	Ratio DiffLib Top 5	DiffLib Top 10	Ratio DiffLib Top. 10	Tiempo ejecución (s)
Journal of Computational Science	J. Comput. Scien.	0		0				2
Data Mining and Knowledge Discovery	Data Min. Knowl. Discov.	1	0.712	1	0.712	1	0.712	2,53
Applied Intelligence	Appl. Intell.	1	0.667	1	0.667	1	0.667	1,53
Journal of Machine Learning Research	J. Mach. Learn. Res.	1	0.571	1	0.571	1	0.571	2,14
Copenaghen Journal of Asian Studies	Copen. Jo. As. Stud.	1	0.581	1	0.581	1	0.581	2,11
International Journal of Computing	Int. J. Comp.	0		0		1	0.489	1,64
Foundation and Trends in Databases	Found. Tren. Datab.	1	0.593	1	0.593	1	0.593	2,14
Handbook of Computational Economics	Hand. Comp. Econ.	1	0.538	1	0.538	1	0.538	1,99
Mathematicals Models and Computer Simulations	Math. Mod. Comp. Sim.	1	0.523	1	0.523	1	0.523	2,13
Quality Innovation Prosperity	Qual. Innov. Pros.	1	0.638	1	0.638	1	0.638	2,2

Tabla 5: Resultados del algoritmo de Difflib