ULPGC
Universidad de Las Palmas de Gran Canaria
Escuela de Ingeniería Informática

Creación de una base de datos de artículos de la ULPGC, incluidos en Google Scholar, para el entrenamiento de métodos de aprendizaje automático

Trabajo Fin de Título

Grado en Ingeniería Informática

Autor

Carlos Artiles Rodríguez

Tutor

Javier Sánchez Pérez

Resumen

El presente proyecto tiene como objetivo facilitar a las universidades la obtención de artículos académicos desde Google Scholar y determinar, haciendo uso de técnicas de aprendizaje automático, si los artículos pertenecen a dicha universidad o no, con el fin de facilitar su importación.

Para cumplir este objetivo, se ha desarrollado un sistema que descarga artículos académicos desde la plataforma Google Scholar y dado que los artículos pueden estar repetidos, se utilizan diferentes algoritmos para filtrarlos.

Por último, se ha creado una red neuronal sencilla mediante la librería Keras, que permite identificar si los artículos pertenecen a la universidad con una fiabilidad del 80%.

Abstract

The present proyect aims to help universities obtain academic articles from Google Scholar and to determine, using Machine Learning techniques, if the articles belong to the university or not, in order to enable their upload.

For this purpose, a system have been develop to facilitate the download of academic articles from the Google Scholar platafform, and since articles might be duplicated, different algorithms are used to filter them.

Finally, a simple neural network has been created using the Keras library, which can determine if the articles correspond to the university with a reliability of 80%.

Índice de contenido

Índice de Figuras

Índice de Tablas

Índice de Ecuaciones

Capítulo 1

Introducción

La producción científica de las universidades es un elemento fundamental para medir la capacidad de investigación que tienen estas, siendo un factor determinante de cara a evaluar su relevancia a nivel internacional. Con el objetivo de preservar y divulgar esta producción generada por los investigadores, las instituciones disponen de un sitio web que almacena en formato digital, todos los recursos académicos que generan. A estos sitios web se les conoce como repositorios institucionales. En el caso de la Universidad de Las Palmas de Gran Canaria (ULPGC) el repositorio que emplea desde marzo de 2020 es accedaCRIS.

Para las universidades, poder registrar por sí mismas toda la producción científica de sus investigadores resulta bastante complicado y trabajoso, por lo que suelen recurrir a las bases de datos académicas como Scopus, Web of Science (WoS) o Dialnet. Cuando los investigadores de estas instituciones publican sus trabajos de investigación en revistas científicas o presentándolos en congresos, sus documentos en forma de artículos o actas son indexados directamente a las bases de datos académicas. Este proceso no se llevará a cabo si la revista o congreso no están indexadas o no lo estaban en el momento de la publicación, de modo que las bases de datos académicas no disponen de todos los documentos científicos de los investigadores de una institución.

Por esto motivo, las universidades suelen recurrir a la plataforma Google Scholar, que es uno de los motores de búsqueda académicos más usados y contiene una mayor cantidad de artículos que las bases de datos académicas, al hacer uso de *crawlers* o rastreadores, identificando sistemáticamente sitios web concretos para extraer su contenido. Esta plataforma tiene una serie de inconvenientes al no controlar correctamente todos los artículos que indexa, pudiéndose encontrar artículos duplicados o que contengan datos erróneos. Tampoco dispone de una API para acceder a todo el material que recopila, por lo que cuando los miembros del departamento de bibliotecas de las universidades tienen que acceder a los datos que se encuentran en Google Scholar supone un gran esfuerzo para ellos y pierden mucho tiempo.

Aquí es donde este proyecto adquiere protagonismo, ya que tiene como objetivo principal la creación de una base de datos que permita entrenar una red neuronal para conseguir con ello, automatizar la subida de artículos desde Google Scholar a accedaCRIS.

En consecuencia, se ha desarrollado un módulo que descarga los datos sobre los investigadores relacionados con la ULPGC y sus artículos desde Google Scholar. Para conseguir realizar este procedimiento, se hace uso de técnicas de *web scraping*, que permiten extraer datos de una página web a partir de su código fuente. Como puede haber artículos repetidos, estos se filtran empleando algoritmos que miden el grado de similitud entre cadenas de texto, y una vez filtrados, se comparan con los artículos que se encuentran en accedaCRIS. De esta comparación se obtiene un conjunto de artículos cuya probabilidad de que pertenezcan a la ULPGC es alta, por

lo que entrenando a la red neuronal con este conjunto de datos, esta aprende a diferenciar cuáles no están en accedaCRIS, y cuales sí deberían estar.

1.1 Objetivos iniciales

El fin principal de este proyecto es la creación de un conjunto de datos que contenga artículos académicos de una institución, obtenidos a partir de Google Scholar, y determinar si esos artículos pertenecen o no a la institución, con la finalidad de facilitar su importación.

Para cumplir con este objetivo principal, se ha comenzado por desarrollar un módulo que obtiene de Google Scholar, tanto a los investigadores relacionados con la institución, como todos sus artículos de forma automática, haciendo uso de técnicas de *web scraping*.

Como Google Scholar asigna muchos artículos a los investigadores de las instituciones que no son correctos, se realiza un proceso de filtrado de artículos con ayuda de la librería `Python Record Linkage Toolkit`. También se usa esta librería para relacionar los artículos de Google Scholar con los de la institución, obteniendo un conjunto de datos con artículos que sabemos que corresponden a la institución.

Por último y como objetivo adicional, se ha creado un clasificador binario que identifica si los artículos procedentes de Google Scholar forman parte de la institución.

Junto a lo anteriormente expuesto, como meta adicional se ha buscado la adquisición de nuevos conocimientos que no se han trabajado durante la carrera, al utilizar nuevas tecnologías nunca empleadas. Con esto se consigue que el alumno sea autodidacta y este formándose continuamente, cualidades importantes para cualquier ingeniero informático.

1.2 Justificación de las competencias especificas cubiertas

Las competencias específicas que se han cubierto durante el desarrollo de este proyecto son las siguientes:

IS01 - Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la ingeniería del software.

Esta competencia está justificada al seguir la filosofía “Clean Code” que facilita el entendimiento del código y permite que pueda ser modificado sin ningún inconveniente.

IS03 - Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

Esta competencia queda cubierta gracias a toda la documentación disponible sobre las diferentes tecnologías que se han usado en este proyecto. Dichas tecnologías incluyen librerías como Pandas, Python Record Linkage Toolkit o Keras de las que se hablara en su apartado correspondiente.

IS04 - Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

Esta competencia se justifica al identificar el problema existente (dificultad a la hora de obtener artículos académicos y subirlos al sistema) y aportar una solución al mismo.

1.3 Aportaciones al entorno

Poder automáticamente obtener un gran número de artículos, clasificarlos y prepararlos para su posterior importación tendrá un impacto beneficioso en el portal de investigación de cualquier institución.

Con esto se consigue, por un lado, que el personal de las bibliotecas universitarias, que son los que normalmente se ocupan de esta tarea, dejen de usar el método tradicional, ya que es bastante lento y, por otro, mantener actualizada la base de datos de la institución, añadiendo continuamente material académico.

Hay que añadir que el contenido de este proyecto pueda ser utilizado de forma abierta para todo aquel que este interado, al estar publicado en la plataforma accedaCRIS perteneciente a la ULPGC, la cual cumple con la directiva europea *Open Access* [1].

1.4 Organización del documento

A continuación, se expondrá un breve resumen de cada uno de los posteriores capítulos que contiene este documento, los cuales son:

- En el Capítulo 2, se explicará la importancia que tienen los repositorios institucionales en el ámbito científico, los programas que estos utilizan para gestionar la producción científica y las bases de datos académicas de las que se obtiene esta producción.
- En el Capítulo 3, se realizará un análisis de las plataformas Google Scholar y accedaCRIS, indicando sus características importantes y sus inconvenientes.
- En el Capítulo 4, se expondrá en detalle el desarrollo del proyecto, las partes en las que se ha dividido, problemas que han surgido durante su realización y las soluciones empleadas.
- En el Capítulos 5, se detallará la metodología y las herramientas utilizadas durante el desarrollo del proyecto.
- En el Capítulo 6, se expresarán las conclusiones que se han alcanzado y los trabajos futuros que se podrían desempeñar a partir de este proyecto.

Capítulo 2

Metodología de trabajo y tecnologías utilizadas

En este capítulo se hablará sobre la metodología empleada en el desarrollo del proyecto, junto a la planificación inicial y cambios que se le realizaron a la misma, y se explicarán las diferentes tecnologías y herramientas utilizadas para cumplir los objetivos del proyecto.

2.1 Metodología de trabajo, planificación inicial y cambios realizados

2.1.1 Metodología de trabajo

Se ha optado por seguir una metodología incremental para el desarrollo de este proyecto al ser la que más se adecúa para este trabajo.

En la metodología incremental las tareas están divididas en iteraciones, en las cuales se busca completar objetivos específicos consiguiendo que el producto muestre una mejora con respecto a la iteración anterior. Estos objetivos no son independientes entre sí y están vinculados de manera que cada uno, suponga un avance respecto al anterior.

Una de las principales ventajas de esta metodología es que gracias a que las iteraciones son pequeñas, se pueden manejar de forma sencilla las tareas de cada iteración y adaptarse a cambios o modificaciones que surjan durante el desarrollo.

2.1.2 Planificación inicial

Teniendo en cuenta la metodología empleada, la planificación inicial del proyecto queda representada en la Tabla 1:

Las primeras horas invertidas en el proyecto se dedicaron a analizar la estructura de los datos que se encuentran en Google Sholar relacionados con los académicos y sus artículos. Al realizar esta inversión de tiempo resulto más sencillo poder identificar los datos relevantes que se obtendrían más delante

En el caso de accedaCRIS, se le facilito al alumno un excel con todas las publicaciones para su análisis lo que facilito la comprensión de los datos. Como no se disponía de los conocimientos necesarios sobre las diferentes librerías que se utilizarían, se le dedico tiempo a su aprendizaje y manejo.

Después de esta fase de análisis, se comenzó el desarrollo del proyecto donde, como comentamos anteriormente, cada tarea se convertiría en una iteración y en cada una obtendríamos un incremento que sería necesario para las siguientes iteraciones. Hay que tener en cuenta que, aunque en la Tabla 1 exista una fase de validación y su correspondiente tarea, la validación de los métodos implementados se realiza al acabar cada iteración siguiendo la metodología incremental.

Tabla 2.1: Plan de trabajo inicial

Fases	Duración estimada (horas)	Tareas
Estudio previo / Análisis	50	Tarea 1.1: Análisis de los datos de la ULPGC contenidos en Google Scholar
		Tarea 1.2: Análisis de las publicaciones en accedaCRIS
		Tarea 1.3: Estudio de la documentación sobre la librería Pandas
		Tarea 1.4: Estudio de la documentación sobre la librería Keras
Diseño / Desarrollo / Implementación	160	Tarea 2.1: Desarrollo de funciones para descarga de datos desde Google Schoolar
		Tarea 2.2: Creación de un dataset de publicaciones de Google Schoolar y accedaCRIS
		Tarea 2.3: Desarrollo de un módulo para establecer correspondencias entre publicaciones de Google Schoolar y las de accedaCRIS
		Tarea 2.4: Anotación semiautomática de publicaciones de Google Schoolar
		Tarea 2.5: Creación de un prototipo de red neuronal simple para probar los datos
Evaluación / Validación / Prueba	60	Tarea 3.1: Evaluación de los métodos implementados
		Tarea 3.2: Análisis del rendimiento de la red neurona
Documentación / Presentación	30	Tarea 4.1: Revisión de documentación producida
		Tarea 4.2: Redacción de la memoria y preparación para la presentación

2.1.3 Cambios en la planificación

Durante la realización del proyecto se identificaron nuevas tareas como, por ejemplo, estudiar la documentación sobre la librería `Python Record Linkage Toolkit` que facilitaría la comparación de artículos académicos. También se ajustaron los tiempos de la fase de desarrollo y documentación. Estos cambios quedan reflejados en la Tabla 2.

Tabla 2.2: Plan de trabajo final

Fases	Duración estimada (horas)	Tareas
Estudio previo / Análisis	60	Tarea 1.1: Análisis de los datos de la ULPGC contenidos en Google Scholar
		Tarea 1.2: Análisis de las publicaciones en accedaCRIS
		Tarea 1.3: Estudio de la documentación sobre la librería Pandas
		Tarea 1.4: Estudio de la documentación sobre la librería Keras
Diseño / Desarrollo / Implementación	185	Tarea 2.1: Desarrollo de funciones para descarga de datos desde Google Schoolar
		Tarea 2.2: Creación de un dataset de publicaciones de Google Schoolar y accedaCRIS
		Tarea 2.3: Desarrollo de un módulo para establecer correspondencias entre publicaciones de Google Schoolar y las de accedaCRIS
		Tarea 2.4: Anotación semiautomática de publicaciones de Google Schoolar
		Tarea 2.5: Creación de un prototipo de red neuronal simple para probar los datos
Evaluación / Validación / Prueba	60	Tarea 3.1: Evaluación de los métodos implementados
		Tarea 3.2: Análisis del rendimiento de la red neurona
Documentación / Presentación	40	Tarea 4.1: Revisión de documentación producida
		Tarea 4.2: Redacción de la memoria y preparación para la presentación

2.2 Tecnologías empleadas

Las tecnologías y herramientas utilizadas para el desarrollo del proyecto se describen a continuación:

2.2.1 Entorno de desarrollo

- **Sublime Text:** es un editor de código y texto, que da soporte a muchos lenguajes de programación y lenguajes de marcas. Permite a los usuarios añadir nuevas funcionalidades mediante *plugins* y tiene una licencia de software libre por lo que se puede descargar y usar de forma gratuita.

2.2.2 Lenguaje de programación

- **Python:** Es un lenguaje de programación multiplataforma y de código abierto que debe su éxito a su fácil uso.

  > "*Python es ideal para trabajar con grandes volúmenes de datos ya que, el ser multiplataforma, favorece su extracción y procesamiento, por eso lo eligen las empresas de Big Data*". [2]

  Debido a que cada vez más se está haciendo uso de técnicas de aprendizaje automático, hay muchos desarrolladores que están interesados en aprender a programar en Python.

  "*El carácter exploratorio del aprendizaje automático se ajusta a la perfección a Python, así nos podemos encontrar librerías como Keras, PyBrain o scikit-learn para realizar tareas de clasificaciones, regresión, clustering, preprocesamiento o generación de modelos de algoritmos"*. [3]

2.2.3 Librerías

- **Pandas:** es una biblioteca que se usa en la manipulación y análisis de datos. Debido a que Pandas se encarga de preparar los datos para su comparación, gestionar campos vacíos y agrupar conjuntos específicos de datos, es bastante utilizada por los programadores para procesar datos a alto nivel en Python.

  La estructura básica con la que trabaja es con lo que se conoce *dataframe*, que es una colección de datos ordenados por columnas y se comporta de manera similar a una tabla SQL, disponiendo de tipos y nombres.

- **Python Record Linkage Toolkit:** es una librería que permite relacionar registros que se encuentren en una colección da datos. Cuenta con métodos para el indexado y filtrado de datos, y funciones para comparar registros por lo que es una librería bastante usada.

También hace uso de la librería Pandas al hacer que el proceso de relacionar registros sea más sencillo y rápido.

- **Keras:** es una biblioteca de código abierto basada en el lenguaje de programación Python y usada en el desarrollo de redes neuronales. Puede ejecutarse en diferentes librerías usadas para el aprendizaje automático como, por ejemplo, TensorFlow, Microsoft Cognitive Toolkit o Theano. Se usa para la experimentación en el aprendizaje profundo, ya que está diseñada para ello. Entré sus características se destaca del resto en ser amigable para el usuario, modular y extensible.

Capítulo 3

Estado del arte

En los últimos años, los repositorios institucionales están cobrando una mayor importancia para la institucionas al ser un medio con el que pueden preservar y difundir su producción científica. Con el fin de desarrollar estos repositorios se utilizan diversos utilizan diversos programas, permitiéndoles estos también, organizar los recursos en base a diferentes ontologías y crear perfiles de los investigadores teniendo en cuenta diferentes criterios como, por ejemplo, su campo de investigación. Estos programas se clasifican en: los de código abierto y los propietarios. Los programas de código abierto cuentan con la ventaja de que no hay que pagar una licencia y permiten la aceptación de nuevos estándares para la gestión de los datos, pero el mantenimiento del sistema recae en la institución. Por otro lado, los programas propietarios son aquellos que pertenecen a una empresa y por la tanto requieren pagar una suscripción, pero esta se compromete a mantener el sistema.

En la actualidad existen diferentes sistemas o plataformas que recogen una gran cantidad de recursos académicos como Google Scholar, Scopus, ResearchGate entre otras. Estos sistemas, no solo permiten que el usuario medio pueda acceder a toda la información proporcionada por investigadores académicos, sino que también, ayudan a los investigadores a conseguir mayor visibilidad a través de las citaciones de sus artículos, tesis, libros, etc. Logrando esta visibilidad, un investigador consigue que su trabajo sea conocido y aumenten sus posibilidades de obtener el éxito en su campo académico.

La mayoría de estos sistemas se consideran bases de datos académicas, siendo algunos una base de datos en sí como en el caso de Scopus y otras adquieren sus recursos académicos al ser plataformas en línea que facilitan el acceso a otras bases de datos como, por ejemplo, Web of Science. En el caso de Google Scholar, obtiene los recursos académicos haciendo uso de *crawlers* o rastreadores [4] que son programas que acceden a páginas concretas extrayendo el contenido de estas haciendo uso de técnicas de *web scraping*. Aunque Google Scholar no cumple esto, la mayoría de estas bases de datos académicas disponen de un API que les permite a las instituciones insertar el material académico de forma cómoda en sus repositorios institucionales.

En el mercado, existen distintas APIs o programas que mediante técnicas de *web scraping*, permiten obtener información de las páginas web. Lo más común, es que extraigan datos de las páginas de resultados de los buscadores, también conocidas como SERP [5]. Entre estas aplicaciones, destaca SerpApi, que puede extraer los datos tanto de los autores como de su material académico desde Google Scholar.

Actualmente, al menos de forma pública, no existen sistemas que se encarguen de extraer datos sobre los artículos académicos de diferentes plataformas y que, además, realicen la tarea de eliminación de duplicados y agrupación. Aunque es bastante probable que, de forma privada, haya

universidades que utilizan sistemas similares para recolectar información y aumentar el material académico de sus repositorios institucionales.

Por lo expuesto anteriormente, este proyecto quiere unir ambos procesos (extraer los artículos y filtrarlos) en un único sistema, consiguiendo acceder a la gran cantidad de material académico que ofrecen estas bases de datos y tratar dicha información con el objetivo de entrenar una red neuronal, que permita automatizar la subida de este material académico en las instituciones.

A continuación, se exponen algunos sistemas que recopilan material académico, dividiéndolos en tres secciones: redes sociales académicas, bases de datos académicas y repositorios institucionales. También se explicará en que consiste la librería SerpApi y se hablara sobre los diferentes programas que usan los repositorios institucionales para gestionar y obtener material académico.

3.1 Redes sociales académicas

Las redes sociales ya forman parte de nuestro día a día y con ellas podemos relacionarnos con otras personas, compartir opiniones, etc. También nos pueden ayudar en el ámbito profesional al servirnos como una carta de presentación.

Para los investigadores académicos, las redes sociales les permiten relacionarse con compañeros, buscar nueva información y difundir sus ideas con la comunidad científica.

3.1.1 ResearchGate

Es una red social gratuita, que permite seguir y ser seguido por otros investigadores con el objetivo de compartir conocimiento. Posee características similares al resto de plataformas como un motor de búsqueda semántica que nos permite navegar por diferentes recursos o poder obtener las estadísticas y métricas sobre el uso de nuestras publicaciones. Al ser una red social dispone de características específicas como que los investigadores pueden ver el trabajo de otros investigadores y comentar sus investigaciones dentro de la comunidad académica mediante foros.

3.1.2 Academia.edu

Red social de registro gratuito que nos permite contactar con investigadores de todo el mundo. Es posible seguir a otros investigadores para ver sus últimos artículos publicados, si están dando alguno charla, etc. También podemos unirnos a grupos de investigación que tengan que ver sobre nuestro campo de estudio y poder compartir información con otros investigadores con preferencias similares a las nuestras. Academica.edu dispone de una función que nos permite importar contactos a partir de otras redes sociales como Twitter, Facebook, etc.

3.2 Bases de datos académicas

Cuando un investigador académico está realizando un estudio y necesita revisar diferentes recursos académicos, quiere estar seguro de que estos recursos sean lo más fiables posibles. Por este motivo las bases de datos académicas son sistemas bastante socorridos, ya que no solo recogen una gran cantidad de información, sino que también ofrecen seguridad sobre los recursos que estos almacenan. Además, disponen de una API que permite de manera sencilla insertar el material académico en los repositorios de las instituciones.

3.2.1 Scopus

Es una base de datos de la editorial científica Elsevier y es una de las más utilizadas a la hora de buscar información. A diferencia de Google Scholar es una plataforma a la que se accede mediante suscripción por lo que hay que registrarse previamente para poder realizar las búsquedas. Su objetivo fundamental es la de evaluar la producción científica a partir de parámetros bibliométricos, dando una idea de cuál es la calidad de la producción científica.

Aunque se basa en resúmenes y citas, en algunos casos se da acceso al texto completo de los documentos que incluye. Dispone características similares otras plataformas, pero la más destacable seria poder ver en que revista publicar un artículo académico, gracias a una evaluación que se puede realizar de las revistas de una materia en concreto.

3.2.2 Web of Science

Plataforma en línea propiedad de Clarivate Analytics, que recoge una gran colección de bases de datos de referencias bibliográficas y citas, diseñada para respaldar la investigación científica y académica. Entre las diferentes bases de datos podemos las que se centran en contenidos específicos como BIOSIS, Derwent Innovations Index, FSTA, Medline y bases de datos regionales de diferentes países como SciELO Citation Index o Korea Citation Index.

Permite a los investigadores buscar una amplia cantidad de recursos académicos y utilizar las relaciones entre citas para llegar a material académico relevante y medir el impacto que este tiene. También los investigadores pueden realizar dichas búsquedas en una base de datos o en varias de forma simultánea.

3.3 Software sobre repositorios y librerías

Los *software* para la creación de repositorios institucionales son herramientas que permiten a las instituciones fundar su propia plataforma digital en línea. Para decidir qué software elegir se lleva a cabo un análisis en el cual se tienen en cuenta aspectos como la licencia, interfaz, etc.

3.3.1 DSpace-CRIS

Se trata de un *software* gratuito de código abierto basado en el *framework* de desarrollo DSpace [6], que se encarga de recolectar y administrar datos de investigación de un sistema CRIS. Es compatible con el modelo CERIF, permitiendo modificar y ampliar las entidades que este utiliza. También interactúa con los entornos de identificadores persistentes ORCID.

Se centra en los repositorios institucionales, buscando proporcionar mayor visibilidad de la información recogida. Las instituciones lo implementan junto a sus repositorios institucionales al permitir gestionar estos datos de forma flexible. Además, ofrece funciones útiles a estas instituciones como gráficos de colaboración, indicadores bibliométricos, bibliografías, etc, pudiendo organizar los datos por autores o por departamentos.

3.3.2 VIVO

Es un *software* gratuito de código abierto disponible para todo el mundo y a la vez una ontología para representar la investigación científica. El sistema VIVO [7] combina una ontología, utilizando clases y propiedades, para crear un perfil de los investigadores en base a su campo de trabajo a través de actividades de investigación (becas y proyectos), publicaciones y docencia.

VIVO quiere animar a las instituciones a divulgar sus investigaciones académicas y ver el impacto que este tiene en la comunidad científica. Aunque se desarrolló para ser utilizado únicamente por instituciones educativas, al ganar bastante popularidad, también lo utilizan agencias gubernamentales o bibliotecas con el objetivo de dar a conocer el trabajo de sus académicos. Dispone de una funcionalidad conocida como "buscador de expertos científicos" que ayuda a los profesores, estudiantes y otras personas a encontrar académicos en las instituciones.

VIVO dispone de una gran cantidad de socios que contribuyen en el proyecto como instituciones académicas, agencias federales, colegios profesionales y proveedores de datos entre los que destacan CASRAI, EuroCRIS y ORCID.

3.3.3 Pure

Es un *software* propietario que reúne información académica sobre investigadores, centros de investigación o proyectos [8]. Proporciona una manera fácil de organizar y mostrar las investigaciones, así como, documentar las relaciones entre estas y los investigadores, ofreciendo una mayor colaboración internacional y una mayor visibilidad de la investigación. Pure destaca

el trabajo académico de los investigadores y los departamentos, ayudando a construir redes de colaboración sólidas tanto en la propia institución como en otras instituciones.

La principal característica de Pure es que permite importar, introducir y mantener datos de alta calidad sobre la investigación y otros contenidos, asegurando la fiabilidad de la información proporcionada.

Pure puede importar datos académicos a partir de diferentes fuentes externas como Scopus, Web of Sciences, Mendeley, etc. Antes de importar estos datos, se utiliza el servicio de "Perfeccionamiento de perfiles" para revisar que las listas de publicaciones estén correctamente estructuradas y sin fallos. Además, permite añadir automáticamente las nuevas publicaciones a los perfiles cuando estos estén disponibles.

3.3.4 SerpApi

Es una API de pago orientada a empresas que se encarga de extraer información de diferentes SERP. Como ya se ha mencionado su característica más destacable es que también permite extraer información de Google Scholar con lo que aumenta aún más su valor único frente al resto de aplicaciones del mercado.

Dispone de un *web service*, que permite a los usuarios probar el funcionamiento de la aplicación en tiempo real. Con este servicio los usuarios pueden seleccionar diferentes buscadores con los que realizar búsquedas, añadiendo diferentes parámetros dependiendo del buscador elegido y poder visualizar los datos que se extraen de estas búsquedas.

3.4 Repositorios institucionales

Son plataformas en línea que buscan difundir, almacenar y preservar la producción científica de una institución, con el objetivo de favorecer su visibilidad al resto de la comunidad científica.

3.4.1 FUTUR

Es el portal de investigación científica de los investigadores e investigadoras de la Universidad Politécnica de Cataluña (UPC) y al ser un repositorio institucional busca dar a conocer el material académico de la UPC.

Se actualiza semanalmente gracias a los datos que obtiene de los sistemas de información corporativos como DRAC y UPCommons. También recaba estos datos, a través de la base de datos de patentes INVENES, del portal CORDIS de proyectos europeos y de Altmetric que recoge las citas de las publicaciones a las redes sociales.

3.4.2 eScholarship

Portal de investigación perteneciente a la Universidad de California que permiten a la comunidad científica relacionada con la institución, tener control directo sobre la creación y difusión de su material académico.

Ofrece un servicio denominado eScholarship Publishing que permite usar distintas herramientas de publicación y producción, entre las que se encuentran servicios de consultoría y soporte profesional. También presenta un servicio llamado eScholarship repository, que asegura la preservación y difusión del material académico.

Capítulo 4

Análisis de Google Scholar y accedaCRIS

Para poder cumplir los objetivos de este proyecto, un aspecto fundamental es el de conocer cómo se estructuran los datos tanto en Google Scholar como en accedaCRIS. Este estudio no solo va a ayudar a saber que datos se pueden extraer, sino que también permitirá entender que son cada uno de los datos y como se podrán comparar estos datos entre sí.

En el caso del buscador académico, Google Scholar es una parte importante, ya que la extracción de los datos se llevará a cabo mediante técnicas de *web scraping* y es necesario comprender la estructura de los datos que utiliza este buscador.

En cuanto al repositorio institucional accedaCRIS, no es necesario utilizar técnicas de *web scraping* para la extracción de los datos, ya que se facilitó un archivo con todos los artículos de la plataforma, pero sigue siendo igual de importante entender también como organiza los datos para poder desempeñar con éxito los demás objetivos del proyecto.

Por lo anteriormente mencionado, en este capítulo se explicará en qué consisten las dos plataformas.

4.1 Google Scholar

Es un motor de búsqueda que se especializa en localizar documentos de carácter académico entre los que se incluyen artículos, revistas especializadas, tesis, etc. Para compilar tanta información hace uso de *crawlers* tal y como lo hace el propio buscador de Google, obteniendo estos documentos de repositorios institucionales y organizaciones de carácter académico.

También permite a los investigadores crearse un perfil y realizar un seguimiento de sus artículos junto a las citas de estos. Google Scholar ofrece estadísticas sobre las citas utilizando diferentes métricas (Citas totales, índice-H e índice-I10) y se encarga de actualizarlas automáticamente cuando se encuentren nuevas citas sobre estos artículos. De esta manera, los investigadores consiguen tener mayor control sobre la información de su propio material académico y una mayor visibilidad.

Por las razones anteriormente mencionadas y al ser una plataforma gratuita, es bastante demandada tanto por estudiantes como por investigadores. Los estudiantes disponen de un amplio abanico de información que pueden utilizar en sus trabajos, y aunque los investigadores también la usan por el mismo motivo, una de sus razones es que Google Scholar dispone de más artículos que otras bases de datos académicas como, por ejemplo, Web of Science.

A continuación, se explicarán algunas de las diferentes características que ofrece la plataforma al buscar perfiles de investigadores, mostrando como estos están estructurados.

Además, se comentarán los problemas específicos que pueden surgir a la hora de extraer los datos de la plataforma.

4.1.1 Búsqueda de perfiles académicos

Cuando se buscan perfiles académicos, se muestra una lista de perfiles que contienen datos como el nombre del investigador, institución a la que está afiliado o número total de citas de sus artículos. De manera similar a la búsqueda de artículos, sí esta tiene éxito y los resultados lo permiten, aparecerán los perfiles de los investigadores encontrados de diez en diez, paginados y en orden descendente teniendo en cuenta el número de totales citas que estos tengan.

Haciendo uso de esta función de Google Scholar, se realizará el proceso de extracción de los datos, tanto de los autores relacionados con la ULGC, como el de sus artículos. Aunque se explicará con mayor detalle en el siguiente capítulo, para ejecutar la extracción se utilizará una librería, a la cual se le facilita la URL de esta página. A continuación, la librería realiza una petición al servidor simulando ser un navegador y, con la respuesta del servidor, se obtiene el código fuente de la página, del cual se extraerán los datos.

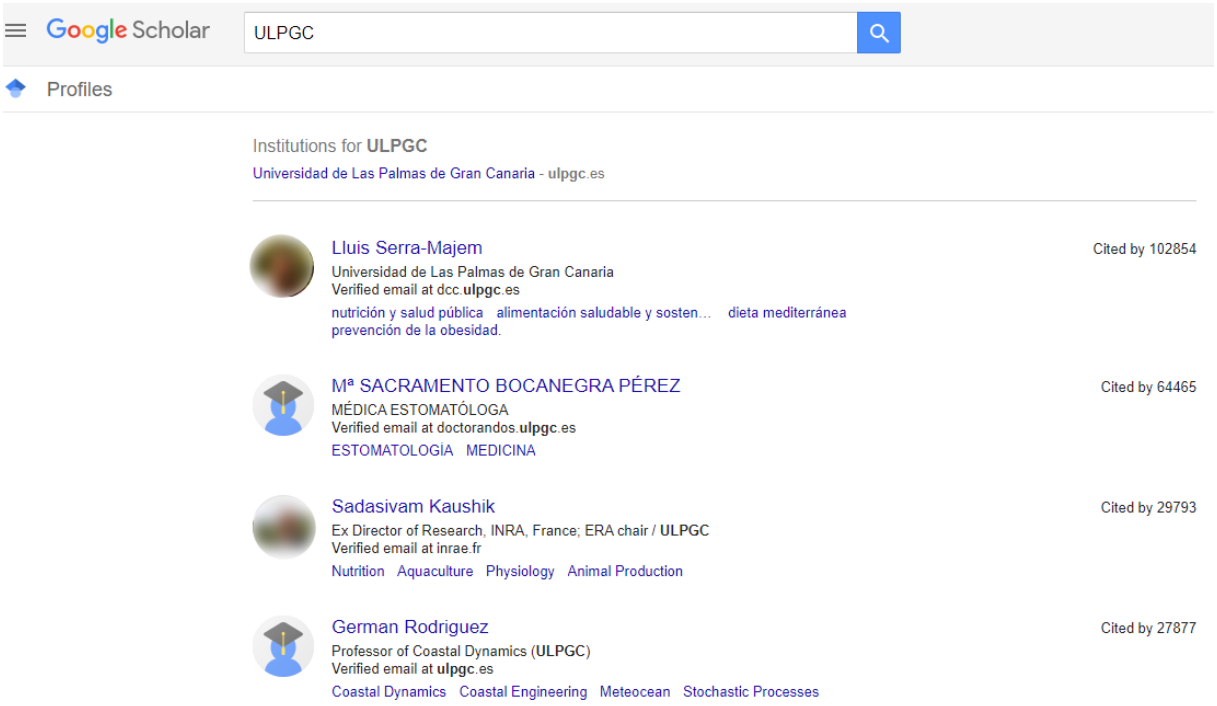


Figura 4.1: Ejemplo de la búsqueda de perfiles académicos en Google Scholar

En la Figura 4.1, se ha realizado la búsqueda de perfiles académicos de la ULPGC. Como se puede ver en dicha figura, el primer resultado que se muestra es la institución “Universidad de las Palmas de Gran Canaria”. Esto se debe a que Google Scholar no solo busca perfiles académicos si no también instituciones que tengan su domino institucional registrado en el sistema. De esta manera, cuando un investigador crea su perfil y en el campo de afiliación pone

alguna de las instituciones, ya sea perteneciendo a una o colaborando con otras, estaría agrupándose dentro de estas.

También se observa en la figura que algunos de los perfiles de investigadores, no deberían estar relacionados con la búsqueda que se ha realizado. El motivo es que Google Scholar a la hora de realizar la búsqueda de un perfil tiene en cuenta los diferentes campos que conforman el perfil del investigador. En este caso al buscar ULPGC, los resultados de la búsqueda muestran perfiles donde aparece ULPGC en las instituciones a las que están afiliados o en el dominio de su correo institucional.

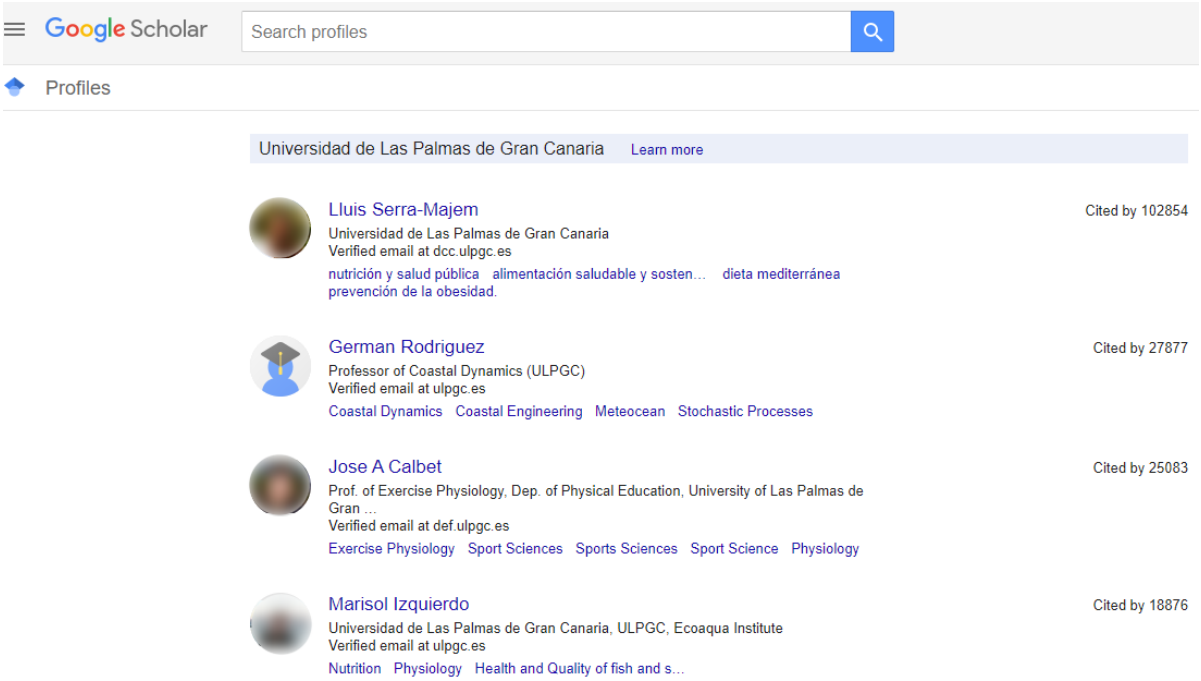


Figura 4.2: Lista de perfiles afiliados a la Universidad de las Palmas de Gran Canaria

Sí se busca Universidad de las Palmas de Gran Canaria o se accede a su agrupación, como puede verse en la Figura 4.2, aparecen aquellos investigadores que en su perfil han puesto la ULPGC como la institución a la que están afiliados. Al comparar los investigadores que aparecen en las Figuras 4.1 y 4.2, puede observarse que no son los mismos, ya que como se explicó anteriormente, hay investigadores que no han puesto en afiliaciones la ULPGC, por lo que no aparecerán asociados a esta institución. Este detalle es algo para tener en cuenta, ya que hay que realizar dos tipos de búsqueda para recabar todos los datos de los investigadores que tienen algún tipo de relación con la ULPGC. Realizando las búsquedas se obtiene que buscando por ULPGC aparecen 728 investigadores y dentro de la institución Universidad de las Palmas de Gran Canaria hay 597 investigadores. Algunos de estos perfiles están repetidos entre las dos búsquedas, por lo que sí se seleccionan únicamente los investigadores que aparecen una vez, se pueden encontrar 767perfiles relacionados con la ULPGC en Google Scholar.

Cuando se accede al perfil de un investigador se pueden diferenciar tres secciones: información personal, artículos y datos bibliométricos.

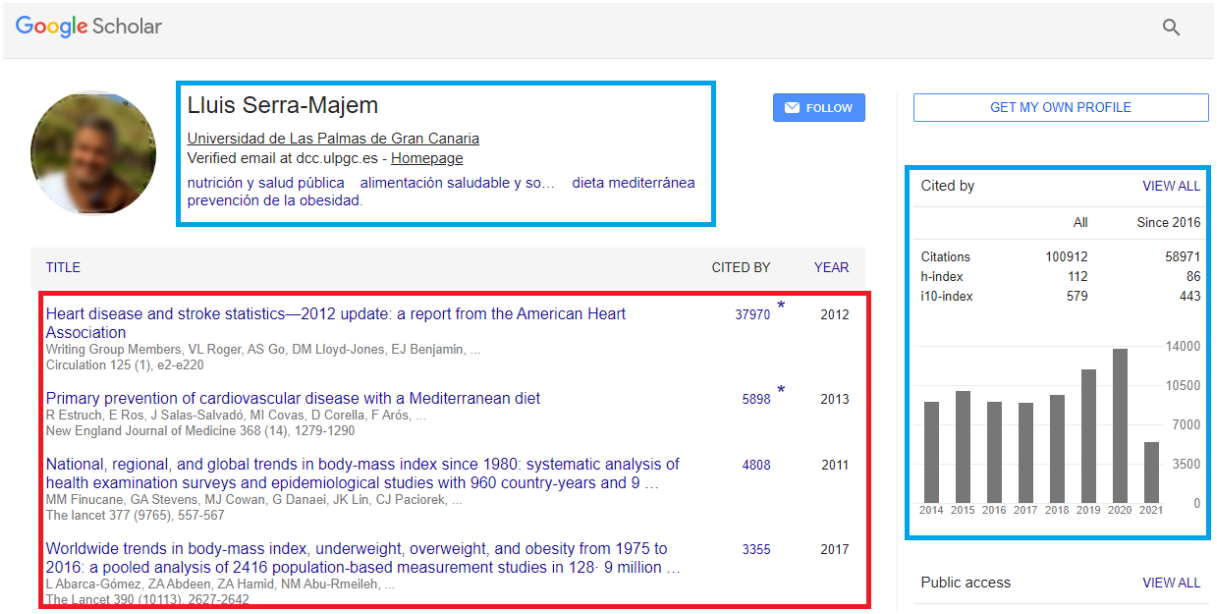


Figura 4.3: Ejemplo del perfil de un investigador en Google Scholar

La sección de información personal se encuentra dentro del rectángulo negro de la Figura 4.3 y contiene los campos que se rellenaron durante el registro del perfil académico:

- El **nombre** del investigador.
- La **institución** a la que este afiliado.
- El **dominio del correo electrónico institucional** que se utilizó durante el registro para disponer de un perfil académico en Google Scholar, junto a un mensaje que confirma que dicho correo esta verificado. En caso de que se utilice un correo electrónico que no se puede comprobar que pertenece a una institución, saldrá un mensaje diciendo que el correo electrónico no está verificado. Google Scholar no hace público el correo electrónico utilizado en el registro, solo el dominio al que pertenece.
- Los **campos de investigación** del investigador.

Google Scholar también le asigna un código a cada investigador, que no tiene ninguna relación con el código ORCID o cualquier otro identificador único. Este código no se muestra de forma directa en la página de cada investigador, pero puede encontrarse en el código fuente de la página o en la URL.

En la sección de artículos se encuentra dentro del rectángulo rojo de la Figura 4.3, se pueden ver una lista con todos los artículos asociados al autor y por defecto están ordenados por el número de citas, aunque también se pueden ordenar por el año de publicación.

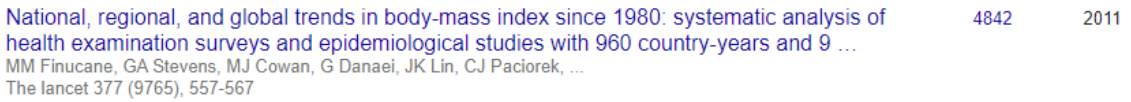


Figura 4.4: Ejemplo de un artículo en Google

Los artículos tal y como se puede ver en la Figura 4.4, están compuesto por:

- **Título** del artículo.
- **Autores** que han participado en dicho artículo.
- **Revista\Libro\Documento** donde se ha publicado el artículo. Puede contener el número de la revista, el número de páginas y la fecha de publicación.
- **Número de citas** que tiene el artículo.
- **Año** en el que se publicó el artículo.

Al igual que con los autores, Google Scholar le asigna a cada artículo un código, el cual tampoco está relacionado con cualquier identificador único, aunque de forma interna seguramente disponga de identificadores como, por ejemplo, el DOI. Este código solo puede ser obtenido del código fuente de la página del investigador o de la página individual de cada artículo de la cual se hablará más adelante. Algo para tener en cuenta a la hora de extraer los datos de los artículos es que, salvo el título y el código interno que se le asigna, el resto de los campos de un artículo pueden encontrarse vacíos.

También otro detalle a considerar es que como se puede ver en la Figura 4.4, el título del artículo y la lista de autores tienen puntos suspensivos indicando que no se puede mostrar el título completamente ni a todos los autores. Para poder acceder a esta información, se puede hacer clic en el título de manera que aparecerá una página individual para cada artículo la cual puede apreciarse en la Figura 4.5.

En esta página individual, se puede ver que aparece la lista completa de autores, pero el título sigue apareciendo incompleto por lo que la única forma de poder saber el nombre completo del artículo sería accediendo a la página de la que Google Scholar extrajo la información del artículo. A esta página se puede acceder haciendo clic en el título o desde el propio enlace que aparece en la esquina superior derecha. Además, en esta página individual aparece más información, como un resumen del artículo, un gráfico que muestra la evolución por año de las citas, los artículos que han citado a este y los artículos que están relacionados.

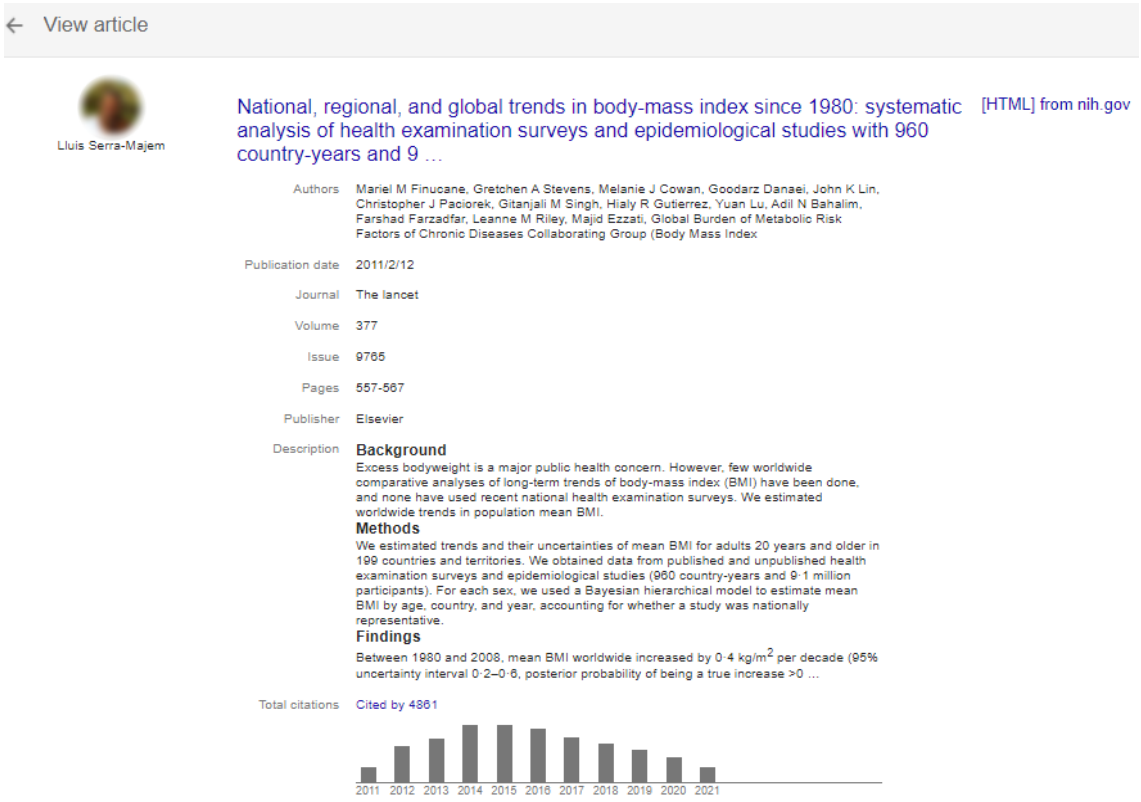


Figura 4.5: Página individual de un artículo en Google Scholar

Tal y como se ha explicado previamente, para poder extraer los datos hay que acceder al código fuente de la página, siendo la mejor opción obtener los datos de la página individual de cada artículo, ya que contiene más información. Sin embargo, esto conlleva un inconveniente y es que cuando se ingresa en una de las páginas individuales, se realiza una petición al servidor y cuando se efectúan muchas peticiones a estas páginas, Google Scholar banea la IP, impidiendo volver a realizar cualquier tipo de petición durante un tiempo.

Teniendo esto en cuenta, se extraerán los datos únicamente de la lista de artículos que muestra el perfil del investigador, aunque estos estén incompletos.

Figura 4.6: Funcionamiento del sistema de duplicados en Google Scholar

Otro de los problemas que acarrea Google Scholar, es que se pueden encontrar artículos repetidos y aunque la plataforma dispone de un sistema que detecta duplicados, este no funciona del todo bien. Cuando el sistema de duplicados encuentra que hay uno o varios artículos iguales, lo indica tachando el número de citas y, sí se hace clic sobre estas, muestra cuales son los artículos que el sistema sugiere que están repetidos. La Figura 4.6, muestra cómo funciona este sistema, donde puede verse claramente que los dos artículos son el mismo, pero al observar cada uno se puede ver que tienen diferentes números de citas, por lo que hasta que el sistema detectó que estos dos artículos estaban repetidos, estuvo asignando las citaciones a cada uno de forma individual. Este ejemplo demuestra que cuando se extraigan los artículos será necesario filtrarlos antes de trabajar con ellos.

Además, hay más problemas que pueden tener los artículos, por ejemplo, que los datos que estos tengan sean incorrectos, que haya artículos asociados a autores erróneamente o que el recuento del número de citas puede ser manipulado [9], siendo estos unos de mayores los problemas que presenta Google Scholar. Durante el desarrollo de este proyecto no se ofrecerá una solución a estos últimos problemas, ya que no se ha encontrado ninguna, pero es algo que se debe considerar antes de realizar el futuro proceso de subida de artículos a la plataforma accedaCRIS.

Por último, el perfil de un investigador cuenta con la sección de datos bibliométricos que se encuentra dentro del rectángulo azul de la Figura 4.3 y que para mayor claridad se ha extraído y puede verse en la Figura 4.7.

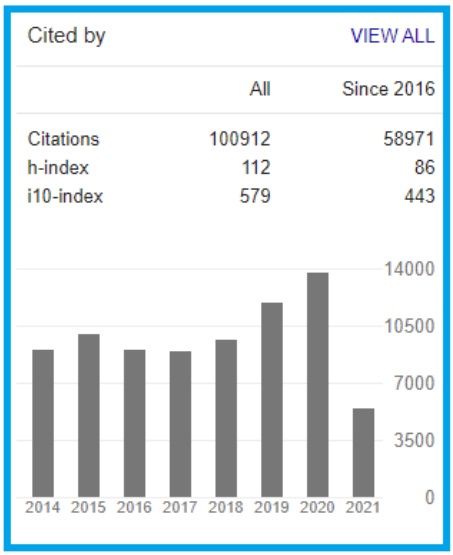


Figura 4.7: Datos bibliométricos de un investigador en Google Scholar

En esta sección, Google Scholar nos muestra el cálculo del número de citas totales y del número de citas del último lustro. Para ello, utiliza diferentes índices entre los que se encuentran el índice H y el índice i10.

- El **índice H** es un sistema de medición de la calidad profesional de los científicos en función del número de artículos que este posea y del número de citas que han recibido sus artículos. Se calcula ordenando los artículos de mayor a menor en función a al número de citas. De esta manera un investigador tiene índice h cuando X artículos disponen de al menos X citas cada uno. Por ejemplo, un índice h igual a 119, significa que un investigador tiene 119 artículos, lo cuales han recibido 119 citas cada uno.

- El **índice i10** también es un sistema de medición de la calidad profesional de los científicos e indica el número de artículos de un investigador que han recibido, al menos, 10 citas.

También permite visualizar un gráfico que contiene el número de citas por años que han recibido todos los artículos permitiendo que los investigadores dispongan de un mayor control de sus artículos.

4.2 accedaCRIS

Se trata de un repositorio institucional en línea que recoge y administra toda la producción científica de la ULPGC, generada por el personal docente e investigador. Desarrollada con la colaboración del Vicerrectorado de Investigación, Innovación y Transferencia, la Biblioteca Universitaria y el Servicio de Informática. Forma parte del sistema CRIS (Current Research Information System), base de datos que almacena y gestiona los datos sobre las investigaciones llevadas a cabo en una institución.

Se ha desarrollado utilizando el software de código abierto DSpace-CRIS, creado por HP y el MIT (Massachussets Institute of Technology). Este software es bastante utilizado por instituciones que utilizan el sistema CRIS al estar enfocado en administrar datos de investigación de un sistema CRIS. Como cualquier repositorio institucional, accedaCRIS busca:

- Dar mayor visibilidad y preservar la producción científica de la institución.
- Mejorar la comunicación e intercambio de información científica entre los investigadores.
- Aumentar la sincronización y administración de los datos, permitiendo que estos estén actualizados.
- Facilitar la exportación de los datos a otros servicios relacionados con la ULPGC

El repositorio obtiene gran parte de la producción científica de los investigadores de la ULPGC a partir de las bases de datos académicas Web of Science (WoS), Scopus y Dialnet, gracias a una API que estas bases de datos proporcionan.

En caso de que haya documentos que sigan sin encontrarse en accedaCRIS, la plataforma permite a los investigadores añadirlos bien a través de la autopublicación, o bien, contactando a través del correo electrónico con el personal administrativo. Los resultados de este proceso pueden verse en la Figura 4.8, que refleja el esfuerzo de los investigadores de la ULPGC, desde los años 2015-2019 y su aportación a la ciencia.

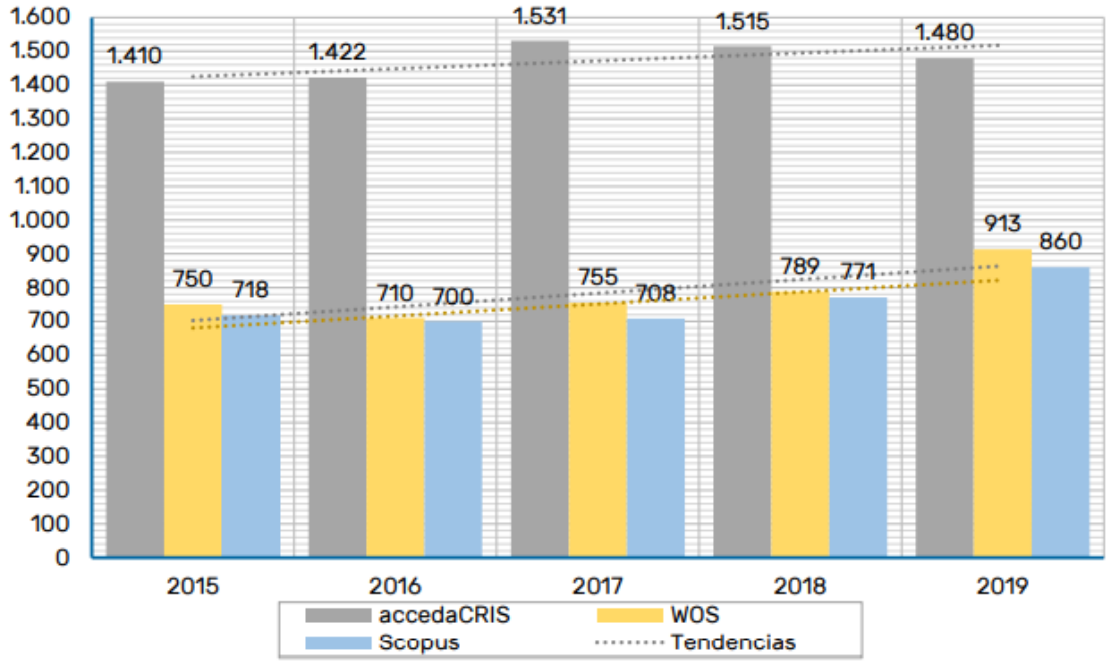


Figura 4.8: Número de publicaciones por base de datos (2015-2019)

(Fuente: Memoria de investigación 2019 [10])

La estructura que sigue la plataforma se dividir en 4 bloques entre los cuales se encuentran:

- **Publicaciones:** se clasifican siguiendo la tipología documental de COAR para asegurar que los metadatos bibliográficos tengan el mismo formato, mejorando la interoperabilidad entre repositorios institucionales que utilicen el sistema CRIS.
- **Personal investigador:** muestra la información personal de los investigadores como, por ejemplo, el nombre completo, afiliaciones, correo electrónico, etc.
- **Organización:** presenta los distintos departamentos e institutos de la ULPGC.
- **Proyectos:** recoge los proyectos, convenios y contratos realizados en la ULPGC.

En los siguientes apartados se explicarán algunos detalles de las secciones de publicaciones y personal investigador, al ser las que más relación tienen con este proyecto.

4.2.1 Publicaciones

En esta sección se encuentran todos los tipos de documentos que produce la ULPGC, los cuales se han clasificado siguiendo el vocabulario controlado de COAR, con el objetivo de mejorar la compatibilidad entre otros repositorios y sistemas CRIS.

Los documentos se dividen en varios subgrupos y disponen de su propia página dentro de la plataforma. Los subgrupos son los siguientes:

- **Publicaciones de investigación:** que incluyen artículos, libros, reseñas, comentarios, entre otros.
- **Publicaciones académicas:** muestra los Proyectos Fin de Carrera, Trabajos final de grado y Trabajos final de máster.
- **Tesis.**
- **Patentes.**
- **Revistas ULPGC:** se pueden visualizar las distintas revistas editadas por la ULPGC.
- **Congresos ULPGC:** recoge las actas de congresos organizados por la ULPGC.
- **Videos ULPGC:** muestra una colección de videos editados por la ULPGC.
- **Datasets ULPGC:** incluye todo el material generado durante una investigación.

Los subgrupos de publicaciones de investigación, publicaciones académicas, tesis y patentes disponen de una gráfica que muestra la producción académica clasificada por años y tipología. La Figura 4.9 presenta el grafico de las publicaciones académicas.

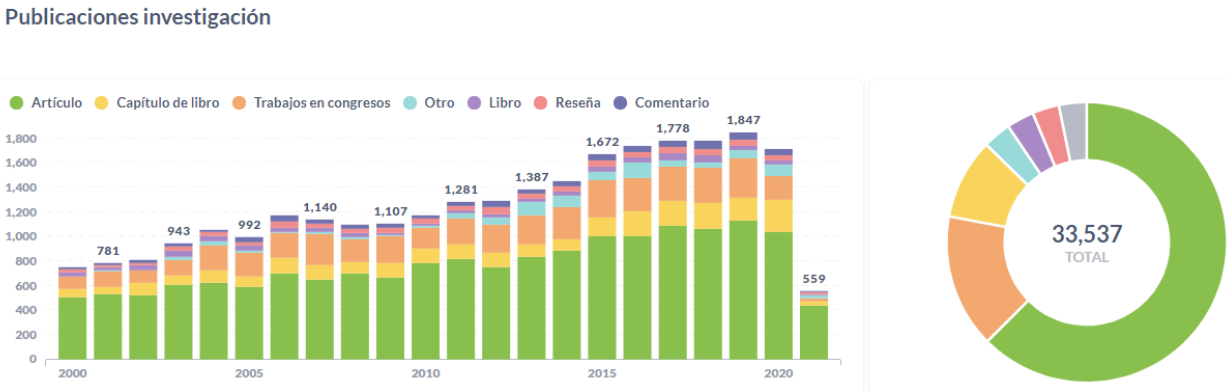


Figura 4.9: Gráfico de las publicaciones de investigación de accedaCRIS

En cada uno de los cuatro subgrupos anteriormente mencionados, se muestra una lista de documentos ordenados por defecto por año de publicación, pero es posible ordenarlos también por título. Además, accedaCRIS ofrece la posibilidad de filtrar estos documentos mediante diferentes opciones que cambian dependiendo del subgrupo, permitiendo localizar mejor los documentos. Dependiendo de si los documentos son una publicación de investigación o una tesis, la información que se muestra sobre estos puede variar.

En la Figura 4.10, que muestra tres documentos del subgrupo publicaciones de investigación, se puede observar que contienen: el título del documento, autores implicados, fecha de publicación, tipo de documento y algunos de sus identificadores únicos en caso de que los tenga.

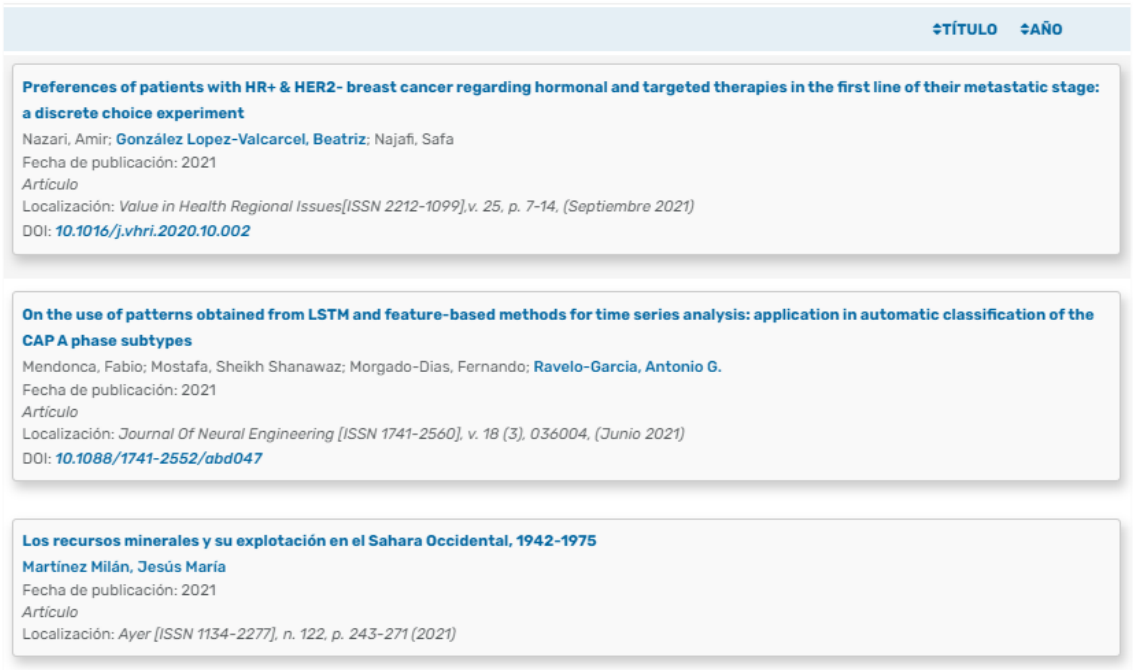


Figura 4.10: Ejemplo de documentos del apartado de publicaciones de investigación de accedaCRIS

Cada documento dispone de su propia página, la cual, contiene más información de la que se muestra en la lista de documentos. También ofrece el propio documento en caso de que este sea público. La página se puede dividir en dos partes: información del artículo y datos bibliográficos. En la parte de información del documento, que se encuentra dentro del rectángulo rojo de la Figura 4.11, se pude ver que un documento en accedaCRIS se organiza de la siguiente manera:

- **Título** del documento.
- **Lista de autores** que han participado en dicho documento.
- **Código UNESCO** que ayuda a identificar el tipo de documento.
- **Palabras clave** que permiten facilitar la búsqueda del documento.
- **Fecha de publicación**.
- **Revista\Libro\Documento** donde se ha publicado el documento.
- **Resumen** del documento.
- Diferentes **identificadores** del documento (Handle, ISSN, DOI).

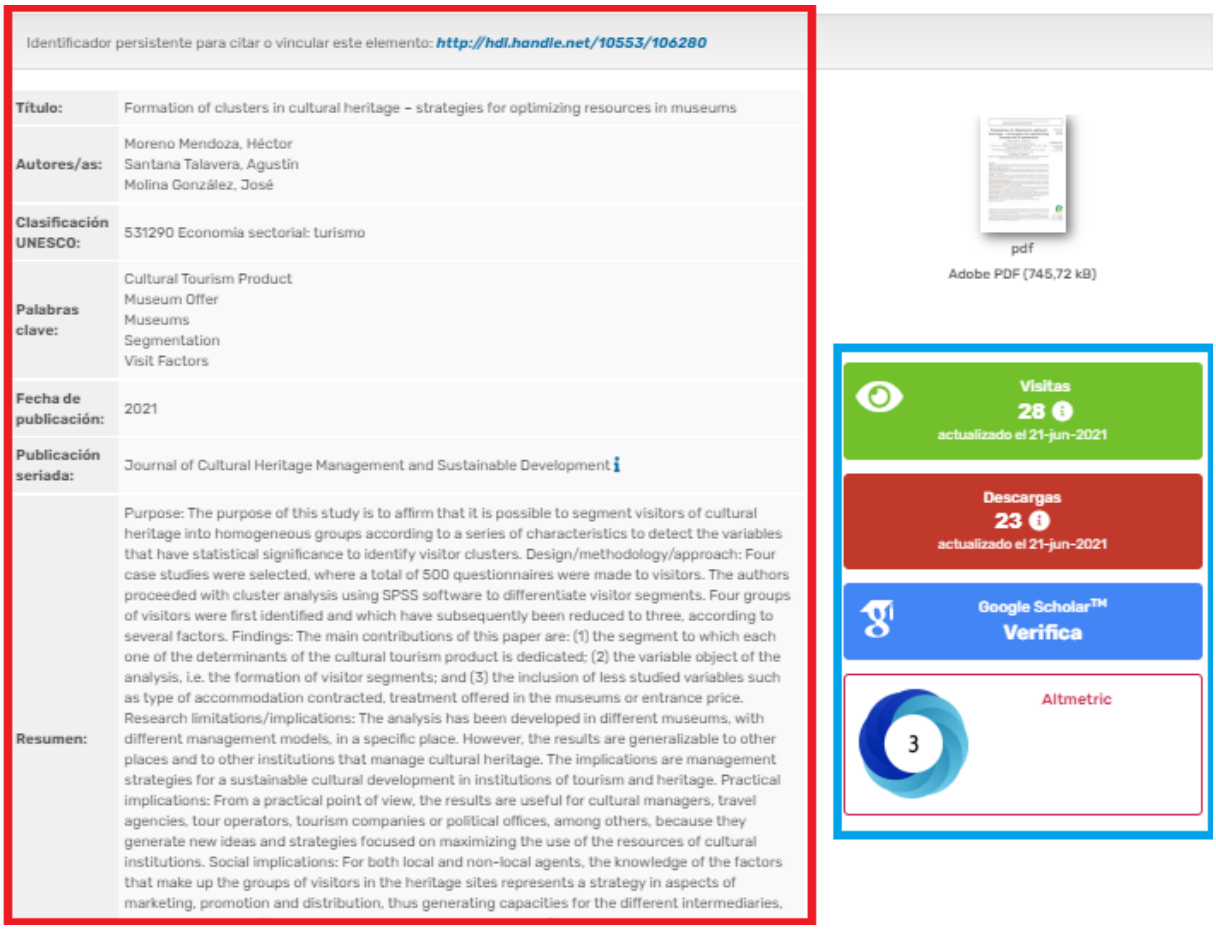


Figura 4.11: Página individual de un documento de accedaCRIS

En cuanto a la sección de datos bibliográficos, la cual se encuentra dentro del rectángulo azul de la Figura 4.11, esta muestra el número de visitas que ha tendido el documento, el número de veces que se ha descargado y el número de veces que se ha compartido. Además, de forma interna accedaCRIS dispone de más información sobre los documentos que no se muestra en la página de estos. En el Excel suministrado por la ULPGC, para el desarrollo de este proyecto, ver Tabla 4.1, se encuentran, a parte de los datos previamente descritos, campos como:

- **ULPGC:** señala si el documento le pertenece a la ULPGC o no.
- **Nº Authors/Local Authors:** indica el número de autores que han participado en el documento y cuáles de ellos son parte de la ULPGC.
- **Journal Volume/Spage/Epage:** muestra el número de la revista donde se publicó el documento, en que página comienza y en cual acaba.
- **SJR Q/SJR Impact:** es un indicador que mide la importancia o influencia científica de un documento en base al número de citas que tenga dicho documento en otros medios. Para medir su índice de impacto se utilizan cuatro cuartiles(Q1,Q2,Q3,Q4), de manera que una revista con un cuartil Q1 tendrá mayor importancia que otra que tenga Q2.
- **JCR Q/JCR Impact:** es otro indicador que determina la importancia de una revista concreta dentro de un mismo campo científico. También hace uso de cuartiles para indicar el índice de impacto que tienen las revistas.

Tabla 4.1: Ejemplo de algunos de los campos extras de los artículos que no se muestra en accedaCRIS

Ulpgc	N Authors	Local Authors	J Volume	Spage	Epage	Sjr Q	Sjr Impact	Jcr Q	Jcr Impact
Sí	4	3	7			Q1	1	Q1	4
Sí	4	4	16	951	963	Q1	1	Q2	3
Sí	5	1	245			Q1	1	Q1	2
Sí	9	1	65	353	365	Q1	1	Q1	4
Sí	9	5	83	121	132	Q2	1	Q3	1
Sí	5	1	76	649	661	Q1	1	Q1	3
Sí	10	4	171	212	230	Q1	2	Q1	4
Sí	5	2	9			Q1	1	Q1	4
Sí	6	1	41	165	176	Q1	1	Q2	2
Sí	4	3							

4.2.2 Personal investigador

En esta sección se encuentra todo aquel investigador perteneciente a la ULPGC que ha publicado y cuenta con algún documento en accedaCRIS. La sección del personal investigador cuenta con un gráfico, que muestra el número total de investigadores y personal docente que disponen de un perfil en acedaCRIS clasificados por categoría. Tal y como se muestra en la Figura 4.12, el número actual de investigadores introducidos en accedaCRIS es 1748.

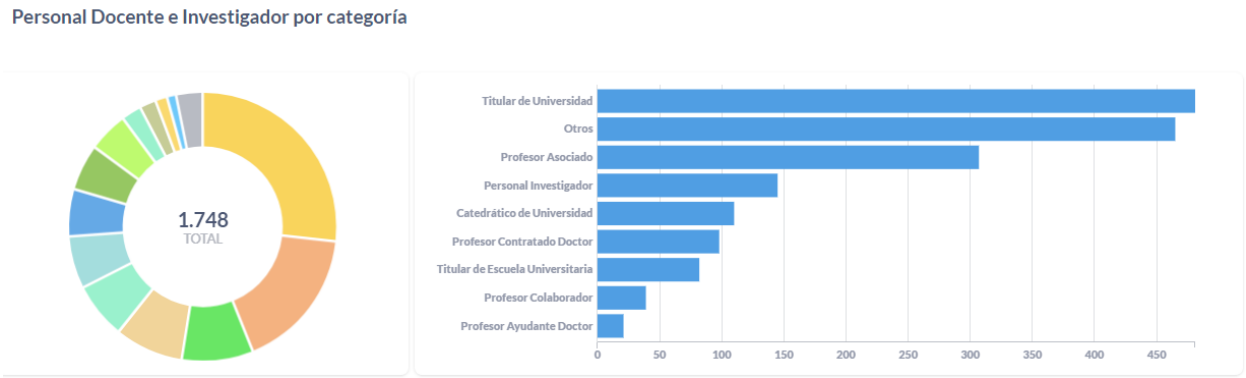


Figura 4.12: Gráfico del personal docente de accedaCRIS clasificados por categoría

Esta sección también muestra una lista de los investigadores, la cual contiene el nombre completo del investigador, su categoría y el departamento al que pertenece. La lista puede ser ordenada por el nombre de forma ascendente (A-Z) o de forma descendente (Z-A), aunque también ofrece la posibilidad de filtrar la lista por categoría o departamento.

Figura 4.13: Lista de investigadores en accedaCRIS

Cada investigador que se encuentre en accedaCRIS dispone de su propio perfil, en el cual se pueden realizar varias acciones como añadir nuevas publicaciones, ver la red de colaboradores y estadísticas sobre su producción científica. Además, contiene diferentes pestañas (Perfil, Publicaciones, Proyectos, etc.) que muestran información sobre el investigador y la producción científica que ha generado en la institución. En la Figura 4.14 se muestra un ejemplo de ello.

Figura 4.14: Ejemplo del perfil de un investigador en accedaCRIS

En cuanto a las diferentes pestañas se encuentra la pestaña Perfil que muestra la información personal del investigador de la siguiente forma:

- **Nombre completo** del investigador.
- **Correo institucional** del investigador.
- **Afiliaciones** del investigador, siendo estas los departamentos, institutos y grupos de investigación a los que pertenece.
- La **categoría** en la que está clasificado el investigador dentro de la ULPGC.

- Diferentes **identificadores** del investigador (ORCID, Scopus ID, ResearcherID, etc.).

Las pestañas de Publicaciones, Proyectos, Tesis, Patentes y TFT, contienen una lista de documentos vinculados al investigador con información sobre los mismos. En la Figura 4.15 se muestra un ejemplo.

Figura 4.15: Ejemplo de las publicaciones asociadas a un investigador en accedaCRIS

Por último, se encuentra la pestaña Indicadores, en la cual los investigadores pueden ver datos bibliométricos sobre sus publicaciones clasificados por año y tipo. Con esta función los investigadores logran tener un mayor control de su producción científica.

Figura 4.16: Ejemplo de datos bibliométricos de un investigador en accedaCRIS

Capítulo 5

Desarrollo del proyecto

En este capítulo se explicarán los procesos realizados para la implementación de este proyecto.

Para comenzar su desarrollo, se ha creado un módulo, que mediante técnicas de *web scraping*, permite descargar los datos de los perfiles de los investigadores relacionados con la ULPGC que se encuentran en Google Scholar, junto a todos los artículos que estos tengan asociados. Con esto se consigue un conjunto de datos de los investigadores y otro conjunto de datos de sus artículos.

A continuación, como en el conjunto de artículos pueden haber duplicados, se ha desarrollado otro módulo que filtra los artículos teniendo en cuenta diferentes campos del conjunto de datos y que, haciendo uso de algoritmos, miden el grado de similitud entre cadenas de texto. Una vez filtrados estos artículos extraídos de Google Scholar, se comparan con artículos que se encuentran en accedaCRIS. De esta manera se obtiene un conjunto de datos con artículos de los que se puede estar casi seguros de que pertenecen a la ULPGC.

Por último, se ha diseñado un modelo de red neuronal sencillo, con el que se pretende que esta pueda aprender a identificar si un artículo pertenece a la ULPGC utilizando el conjunto de datos obtenido en el paso anterior. Por lo tanto, se podrían descargar directamente los artículos de Google Scholar y cuando la red neuronal los clasifique como pertenecientes a la ULPGC prepararlos para su importación en accedaCRIS.

Hay que aclarar que en las imágenes y datos que se expondrán, se usará información relacionada con la ULPGC.

5.1 Creación del módulo de descarga de datos

Esta parte tiene como objetivo obtener de una institución mediante técnicas de *web scraping* [11], los datos de sus académicos y artículos, a partir de Google Scholar.

Al completarla, se obtendrán el archivo **authors-GS.csv** que contiene los datos de los académicos y el archivo **paper-cites-GS.csv** que incluye los datos de los artículos de estos académicos. Además, se explicarán los diferentes problemas que puedan surgir al utilizar técnicas de *web scraping* en Google Scholar.

El siguiente diagrama sirve de guía de los procesos realizados y de los resultados obtenidos:

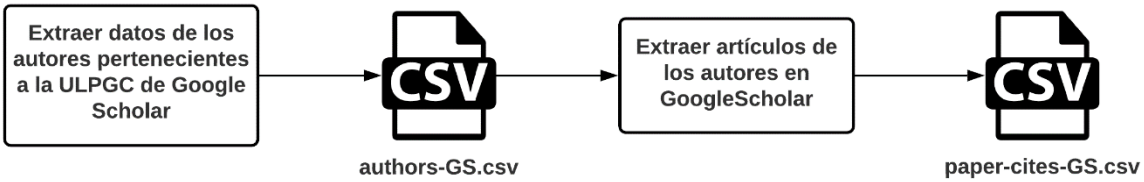


Figura 5.1: Diagrama de bloques de la creación del módulo de descarga de datos

(Fuente: Iconos realizados por Freepik - www.flaticon.com)

5.1.1 Extracción de los datos de un autor

Los datos que nos interesa extraer en esta tarea son: el nombre del investigador, el código de autor con el que identifica Google Scholar a los investigadores y sus datos bibliométricos, aunque estos últimos solo se pueden obtener en el perfil de cada investigador. Como se explicó en su correspondiente capítulo hay que realizar dos tipos de búsquedas para obtener a todos los autores relacionados con la ULPGC. Primero se realizará la búsqueda de perfiles que están afiliados a la Universidad de las Palmas de gran Canaria como se puede ver en la Figura 5.2.

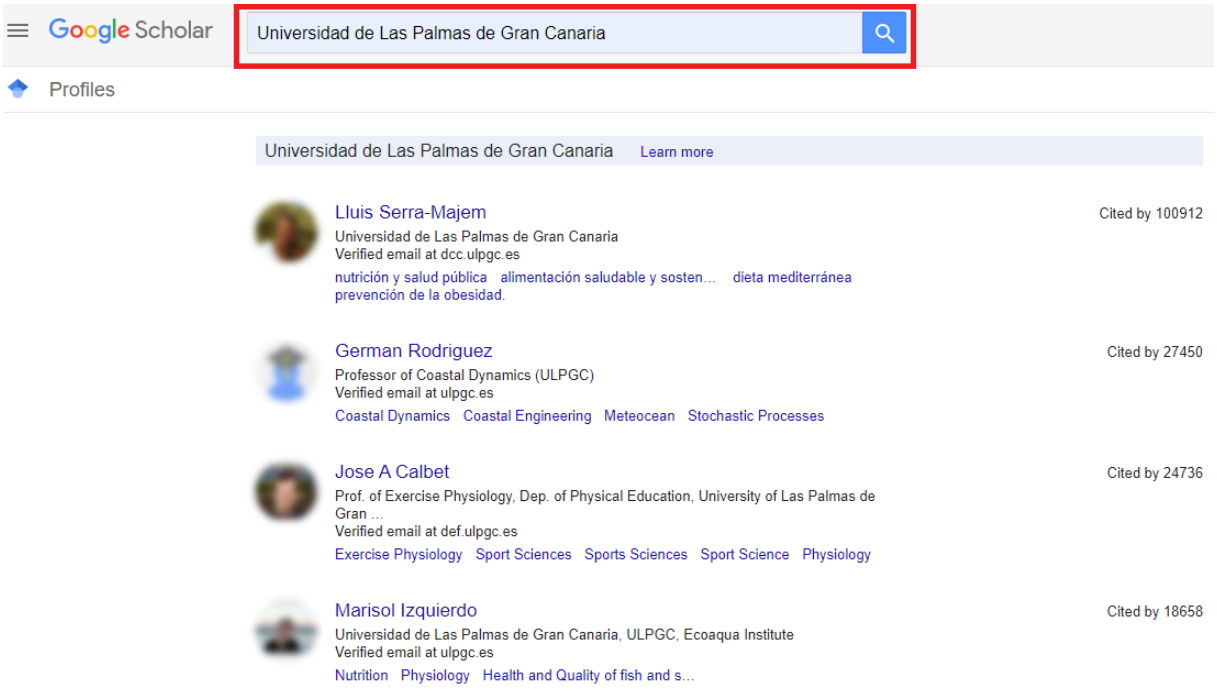


Figura 5.2: Lista de investigadores en Google Scholar

Para obtener estos datos se aplican técnicas de *web scraping* utilizando la librería urllib. A la librería se le proporciona la URL de la búsqueda realizada y esta efectúa una petición al servidor, recibiendo de él un objeto de tipo respuesta. Este objeto, contiene el código fuente de la página y al ser un archivo, se puede leer sin mayor dificultad. De esta manera, podemos tratar el código fuente de la página como una cadena de texto y usar expresiones regulares, para sacar la

información. Con ayuda de la librería `re`, que permite gestionar las expresiones regulares, se usa un patrón para extraer tanto el código del investigador como su nombre. Esto queda reflejado en la Figura 5.3.

```
<div class="gsc_1usr">
        <div class="gs_ai gs_scl gs_ai_chpr">
                <a href="/citations?hl=en&amp;user=nyER5y0AAAAJ" class="gs_ai_pho">
                        <span  class="gs_rimg gs_pp_sm">
                                <img alt="Lluis Serra-Majem" sizes="56px" src="https://scholar.googleusercontent.com/citation
                                user=nyER5y0AAAAJ&amp;citpid=4" srcset="https://scholar.googleusercontent.com/citations?view_
                                user=nyER5y0AAAAJ&amp;citpid=4 56w,https://scholar.googleusercontent.com/citations?view_op=vi
                                citpid=4 128w" width="56" height="53">
                        </span>
                </a>
                <div class="gs_ai_t">
                        <h3 class="gs_ai_name">
                                <a href="/citations?hl=en&amp;user=nyER5y0AAAAJ">Lluis Serra-Majem</a>
                        </h3>
                        <div class="gs_ai_aff">Universidad de Las Palmas de Gran Canaria</div>
                        <div class="gs_ai_eml">Verified email at dcc.ulpgc.es</div>
                        <div class="gs_ai_cby">Cited by 100912</div>
                        <div class="gs_ai_int">
                                <a class="gs_ai_one_int"href="/citations?hl=en&amp;view_op=search_authors&amp mauthors=label:
                                        nutrición y salud pública
                                </a>
                                <a class="gs_ai_one_int" href="/citations?hl=en&amp;view_op=search_authors&amp;mauthors=label
                                        alimentación saludable y sostenible
                                </a>
                                <a class="gs_ai_one_int" href="/citations?hl=en&amp;view_op=search_authors&amp;mauthors=label
                                        dieta mediterránea
                                </a>
                                <a class="gs_ai_one_int" href="/citations?hl=en&amp;view_op=search_authors&amp;mauthors=label
                                        prevención de la obesidad.
                                </a>
                        </div>
                </div>
        </div>
</div>
```

Figura 5.3: Código fuente de la lista de investigadores en Google Scholar

Después de obtener tanto el nombre del investigador como el código de autor, el siguiente paso es el de obtener los datos bibliográficos sobre los investigadores, accediendo al perfil de cada uno de ellos, mediante los códigos de autor previamente extraídos. Para ello se vuelve a usar la librería `urlib`, a la que se le proporciona una ULR fija con cada uno de los códigos de autor, y se obtienen el objeto tipo respuesta que devuelve el servidor de cada uno de los perfiles. A estos objetos que contienen el código fuente se les aplica el mismo proceso usado para extraer el nombre y el código de los autores, de manera que se obtienen las diferentes métricas que usa Google Scholar para las citaciones del material académico de cada investigador. Esto proceso queda reflejado en la Figura 5.4 y la Figura 5.5.

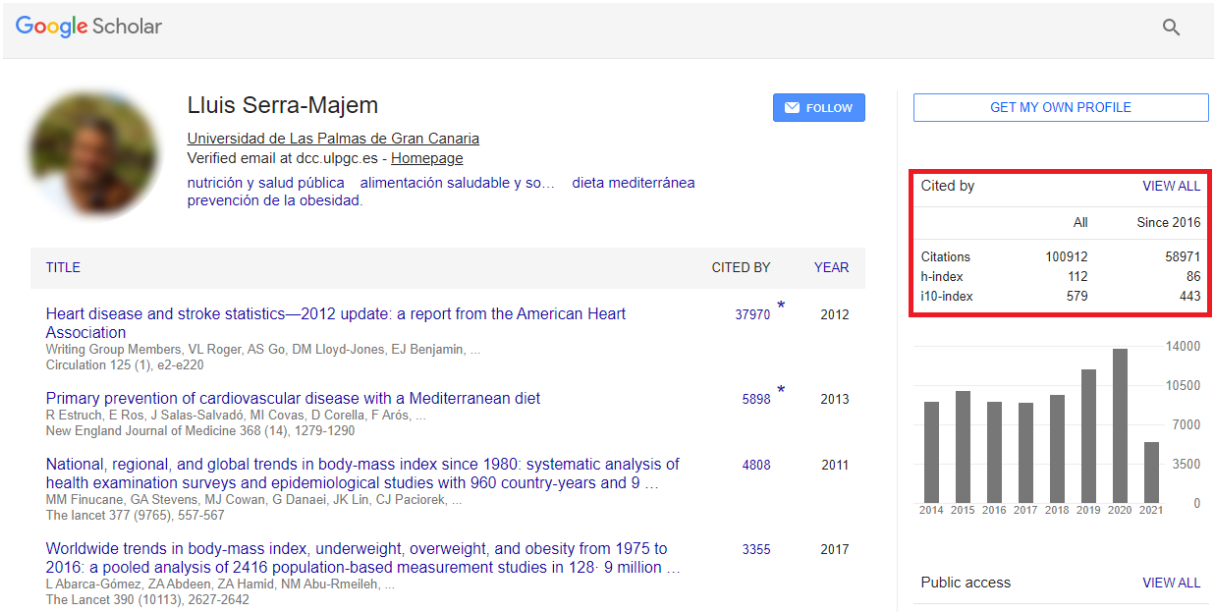


Figura 5.4: Perfil de un investigador en Google Scholar

```
<table id="gsc_rsb_st">
        <thead>
                <tr>
                        <th class="gsc_rsb_sth"></th>
                        <th class="gsc_rsb_sth">All</th>
                        <th class="gsc_rsb_sth">Since 2016</th>
                </tr>
        </thead>
        <tbody>
                <tr>
                        <td class="gsc_rsb_sc1"><a href="javascript:void(0)" class="gsc_rsb_f gs_ibl" title="This is the number o
                        publications. The second column has the &quot;recent&quot; version of this metric which is the number of
                        last 5 years to all publications.">Citations</a></td>
                        <td class="gsc_rsb_std">100912</td>
                        <td class="gsc_rsb_std">58971</td>
                </tr>
                <tr>
                        <td class="gsc_rsb_sc1"><a href="javascript:void(0)" class="gsc_rsb_f gs_ibl" title="h-index is the large
                        h publications have at least h citations. The second column has the &quot;recent&quot; version of this me
                        largest number h such that h publications have at least h new citations in the last 5 years.">h-index</a>
                        </td>
                        <td class="gsc_rsb_std">112</td>
                        <td class="gsc_rsb_std">86</td>
                </tr>
                <tr>
                        <td class="gsc_rsb_sc1"><a href="javascript:void(0)" class="gsc_rsb_f gs_ibl" title="i10-index is the num
                        with at least 10 citations. The second column has the &quot;recent&quot; version of this metric which is
                        publications that have received at least 10 new citations in the last 5 years.">i10-index</a>
                        </td>
                        <td class="gsc_rsb_std">579</td>
                        <td class="gsc_rsb_std">443</td>
                </tr>
        </tbody>
</table>
```

Figura 5.5: Código fuente del perfil de un investigador en Google Scholar

Una vez obtenidos los datos de los autores afiliados a la ULGC, se realiza el mismo proceso, pero efectuando la búsqueda de los autores relacionados con la ULPGC, suministrando a la librería `urlib` la URL de la nueva búsqueda.

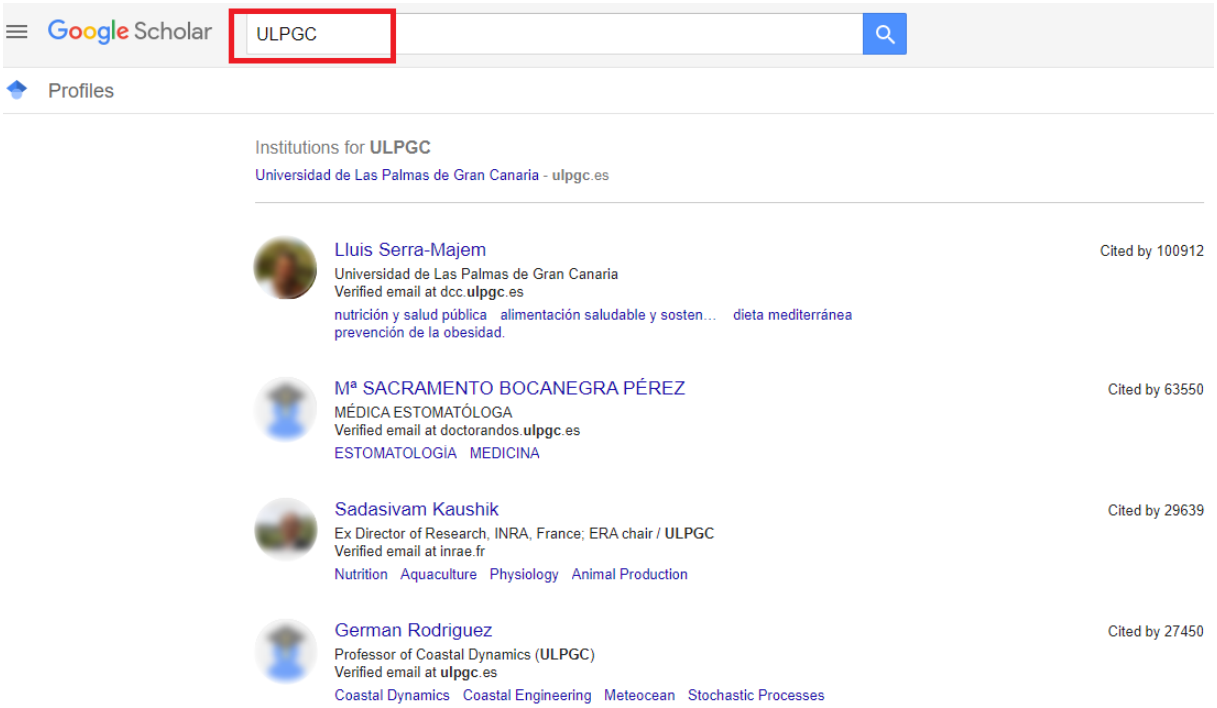


Figura 5.6: Lista de investigadores en Google Scholar (búsqueda alternativa)

Con esto, finaliza el proceso de recolección de datos de un investigador, donde con los nombres y códigos de autor extraídos dentro del rectángulo rojo de la Figura 5.3, y las citas que se encuentran dentro de los rectángulos azules en la Figura 5.5, se obtienen dos conjuntos de datos: uno que contiene los datos de los investigadores que Google Scholar agrupa como pertenecientes a la ULPGC, y el otro, con los datos de los investigadores que están relacionados de alguna forma con la ULPGC. Estos conjuntos contienen:

- **Nombre:** Nombre del investigador.
- **GS-Code:** Código con el que se le identifica en Google Scholar.
- **Citas/Citas(5):** Número total de citas de todas sus publicaciones actuales y del último lustro.
- **h-index/h-index(5)** : Número h de publicaciones que se han citado h veces actualmente y del último lustro.
- **i10-index/ i10-index(5):** Número de publicaciones que se han citado al menos 10 veces actualmente y del último lustro.

Tabla 5.1: Datos extraídos de los investigadores en Google Scholar

GS-Code	Citas	Citas(5)	h-index	h-index(5)	i10-index	i10-index(5)
nyER5y0AAAAJ	105053	62976	114	87	584	446
R6OaE3UAAAAJ	31538	14216	61	42	443	229
fK5WgicAAAAJ	25133	13472	75	52	198	162
DZMTI2UAAAAJ	18909	8159	73	46	204	182
akpWHhIAAAAJ	15351	7158	54	37	361	180
KsuhNAoAAAAJ	12652	5773	43	29	224	104
-h7sZ4wAAAAJ	9220	4268	53	31	126	101
vbdOLdAAAAAJ	8992	4636	25	18	41	31
BILN9dsAAAAJ	8801	3968	48	36	90	85
qmaASSoAAAAJ	7841	3559	40	28	168	88

Hay que añadir que para la explicación se ha usado únicamente la primera página de la búsqueda de perfiles pertenecientes a la ULPGC y realmente el script que extrae los datos, avanza a las siguientes páginas de investigadores, añadiendo a la URL el parámetro correspondiente, hasta obtener todos los investigadores que pertenecen a la ULPGC.

Como entre estos dos conjuntos puede haber investigadores repetidos al encontrarse en las dos búsquedas, se utiliza la función `merge` de la librería Pandas para combinarlos y obtener un único conjunto de datos sin investigadores duplicados.

Por último, a partir del conjunto de datos que no contiene duplicados, se genera un archivo llamado **authors-GS.csv** que será utilizado en el siguiente paso.

5.1.2 Extracción de artículos de un investigador

En este paso se obtiene el material académico de los investigadores pertenecientes o relacionados con la ULPGC. El desarrollo de esta extracción se realiza de forma similar al paso anterior, pero teniendo en cuenta algunos detalles y problemas que surgen en esta parte. Durante la siguiente explicación, se sustituirá material académico por artículos, entendiendo como artículos: libros, tesis, conferencias, etc.

En primer lugar, se accederá al perfil de cada autor, haciendo uso del archivo **authors-GS.csv** obtenido en el paso anterior, puesto que este recoge el código de los investigadores de los cuales nos interesa obtener sus artículos. Con estos códigos y una URL fija, se consigue acceder al perfil de cada autor haciendo uso de la librería urllib. Este proceso queda reflejado en las Figuras 5.7 y 5.8

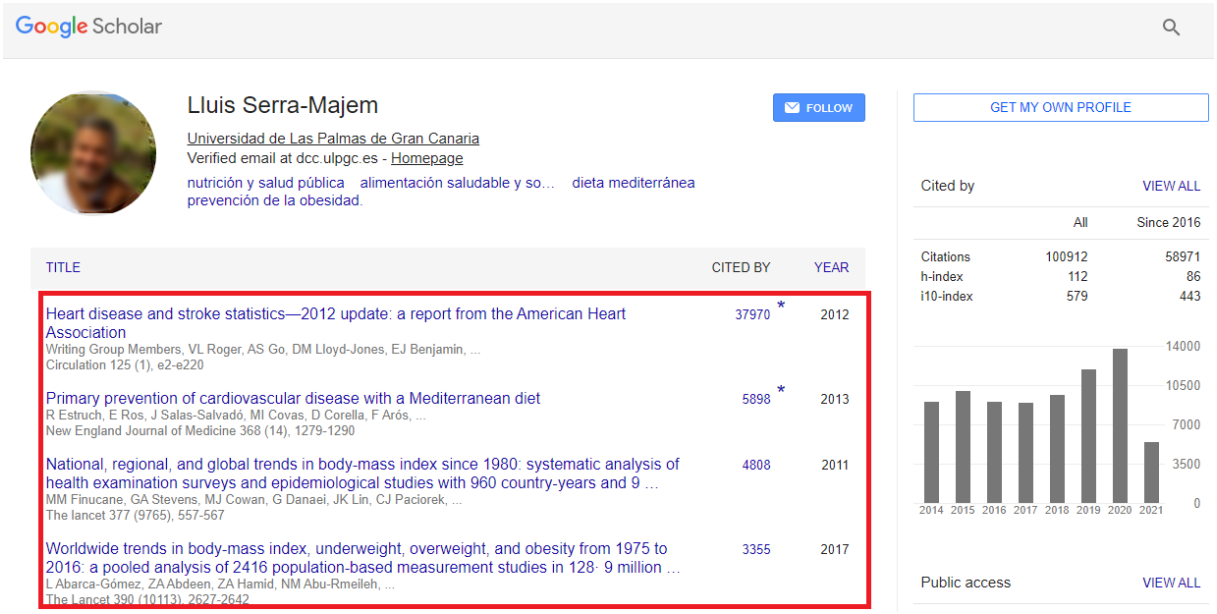


Figura 5.7: Perfil de un investigador en Google Scholar(Artículos)

```
<tr class="gsc_a_tr">
        <td class="gsc_a_t"><a href="/citations?view_op=view_citation&amp;hl=en&amp;user=nyER5y0AAAAJ&amp;
        citation_for_view=nyER5y0AAAAJ:ExNiBuTMO9IC" class="gsc_a_at">

                Heart disease and stroke statistics—2012 update: a report from the American Heart Association</a>

                <div class="gs_gray">Writing Group Members, VL Roger, AS Go, DM Lloyd-Jones, EJ Benjamin, ...</div>

                <div class="gs_gray">Circulation 125 (1), e2-e220<span class="gs_oph">, 2012</span></div>

        </td>
        <td class="gsc_a_c"><a href="https://scholar.google.fr/scholar?oi=bibs&amp;hl=en&amp;cites=1542958399579433181,17
        14543367928605602966,13337470486145177255,12960707862178439566,2965483413014875 10,14100364970459236610,6162116165
        17840597882284810877,9638738354911546971" class="gsc_a_ac gs_ibl">37970</a><span class="gsc_a_m"><a href="javascr
        <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl">2012</span></td>
</tr>
<tr class="gsc_a_tr">
        <td class="gsc_a_t"><a href="/citations?view_op=view_citation&amp;hl=en&amp;user=nyER5y0AAAAJ&amp;
        citation_for_view=nyER5y0AAAAJ:u5HHmVD_uO8C" class="gsc_a_at">

        Primary prevention of cardiovascular disease with a Mediterranean diet</a>

        <div class="gs_gray">R Estruch, E Ros, J Salas-Salvadó, MI Covas, D Corella, F Arós, ...</div>

        <div class="gs_gray">New England Journal of Medicine 368 (14), 1279-1290<span class="gs_oph">, 2013</span></div>
        </td>
        <td class="gsc_a_c"><a href="https://scholar.google.fr/scholar?oi=bibs&amp;hl=en&amp;cites=5996209126120129660,15
        class="gsc_a_ac gs_ibl">5898</a><span class="gsc_a_m"><a href="javascript:void(0)" class="gsc_a_am" data-eid="nyE
        <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl">2013</span></td>
</tr>
```

Figura 5.8: Código fuente de los artículos de un investigador

Los datos que se extraen de los artículos de un investigador son los que se marcan dentro de los rectángulos azules de la Figura 5.8, formando un conjunto de datos con la siguiente información:

- **GS-Code:** código con el que Google Scholar identifica a los investigadores.
- **Art-Code:** código con el que Google Scholar identifica a los artículos.
- **Article:** título del artículo.
- **Authors:** autores que han participado en dicho artículo.
- **Journal:** revista donde se ha publicado el artículo. Puede contener el número de la revista, el número de páginas y la fecha de publicación.
- **Cites:** número de veces que se ha citado el artículo.
- **Year:** año en el que se publicó el artículo.

Tabla 5.2: Datos de los artículos de un investigador en Google Scholar

GS-Code	Art-Code	Article	Authors	Journal	Cites	Year
nyER5y0AAA	nyER5y0AAAAJ:u5HHmVD_uC	Primary pr	R Estruch,	New Engla	5817	2013
nyER5y0AAA	nyER5y0AAAAJ:v6i8RKmR8To	National,	MM Finuc	The lancet	4774	2011
nyER5y0AAA	nyER5y0AAAAJ:kQqwFFzsCTw	Worldwid	L Abarca-G	The Lance	3268	2017
nyER5y0AAA	nyER5y0AAAAJ:qjMakFHDy7s	Mediterra	A Bach-Fa	Public hea	1238	2011
nyER5y0AAA	nyER5y0AAAAJ:2osOgNQ5qM	Food, you	L Serra-Ma	Public hea	949	2004
nyER5y0AAA	nyER5y0AAAAJ:6gSKFiM3Xos	Obesidad	L Serra Ma	Med. clín (	873	2003
nyER5y0AAA	nyER5y0AAAAJ:BrOSOlqYqPU	Medicina	GP Gil, JF	Masson, 2	858	2001
nyER5y0AAA	nyER5y0AAAAJ:YsMSGLbcyi4	A short sc	H Schröde	The Journ	796	2011
nyER5y0AAA	nyER5y0AAAAJ:LNjCCq68lIgC	Dietary α-	G Zhao, TD	The Journ	766	2004
nyER5y0AAA	nyER5y0AAAAJ:u-x6o8ySG0sC	Scientific	L Serra-Ma	Nutrition	705	2006

En el capítulo analizando la plataforma se mencionó que a la hora de extraer los datos había que tener en cuenta algunos detalles como, por ejemplo, que cuando se extraen los datos de los artículos, algunos campos pueden estar incompletos y solo se pueden obtener de forma completa cuando se accede a la página individual del artículo. El inconveniente está en que al realizar esto muchas veces, conlleva un bloqueo de la IP, por lo que no se podrán realizar más peticiones al servidor hasta pasados unos días. Para evitar esto, se ha decidido únicamente coger los datos, aunque estén incompletos y solo acceder a la página del artículo para obtener los datos completamente, cuando se preparen para su importación.

Finalizado el proceso de obtención de artículos, se genera a partir de este conjunto de datos, un archivo con el nombre **paper-cites-GS.csv**, concluyendo así la obtención de datos.

5.1.3. Problemas al utilizar técnicas de web scraping

En el momento de aplicar técnicas de *web scraping* para extraer los datos del código fuente, se pueden encontrar una serie de inconvenientes relacionados con el uso de estas técnicas y con la aparición de información innecesaria que hay que tratar.

Uno de los inconvenientes que se producen tiene que ver con las imágenes vectoriales, que son imágenes que se forman a partir de vectores, permitiendo almacenar más información sin ocupar tanto espacio. Google Scholar las usa para representar fórmulas matemáticas, como por ejemplo raíces cuadradas, pudiéndose observar un esto en la Figura 5.9 y la representación de esta raíz cuadrada en el código fuente se puede ver en la Figura 5.10.

Figura 5.9: Ejemplo de imagen vectorial

```
<td class="gsc_a_t">
        <a href="javascript:void(0)" data-href="/citations?view_op=view_citation&amp;hl=en&amp;user=RBYxSdEAAAAJ&amp;pagesize=100&amp;cit
        RBYxSdEAAAAJ:YXPZ0dOdYS4C" class="gsc_a_at">Measurement of the J/ψ pair production cross-section in pp collisions at s = 13
                <svg class="gs_fsvg" aria-label=" \sqrt{s}=13 " width="62px" height="18px" style="vertical-align:-5px;"><g transform=
                "matrix(0.01700, 0.00000, 0.00000, 0.01700, 0.00000, 13.55749)"><g><path transform="matrix(0.48828, 0.00000, 0.00000,
                -0.48828, 0.00000, -717.49945)" d="M 719 -1954  L 311 -1057  L 190 -1149  Q 184 -1155 176 -1155  Q 167 -1155 157 -1146
                 T 147 -1126  Q 147 -1116 156 -1110  L 399 -926  Q 405 -920 414 -920  Q 427 -920 434 -934  L 801 -1741  L 1671 63  Q
                 1681 82 1706 82  Q 1723 82 1735 70  T 1747 41  Q 1747 31 1745 27  L 793 -1948  Q 780 -1966 760 -1966  H 737  Q 727
                -1966 719 -1954  Z "/><g transform="translate(833.33002, 0.00000)"><g><path transform="scale(0.48828, -0.48828)"
                d="M 178 125  Q 233 31 399 31  Q 471 31 536 55  T 643 129  T 686 248  Q 686 301 648 335  T 555 381  L 444 403  Q 368
                422 319 474  T 270 600  Q 270 691 319 761  T 450 868  T 618 905  Q 711 905 784 860  T 858 729  Q 858 682 831 646  T
                758 610  Q 731 610 711 627  T 692 672  Q 692 696 705 718  T 741 754  T 788 768  Q 770 812 720 832  T 614 852  Q 562
                852 510 831  T 426 769  T 395 674  Q 395 637 421 609  T 485 569  L 604 545  Q 661 533 708 502  T 783 425  T 811 319
                Q 811 243 768 169  T 664 51  Q 555 -23 397 -23  Q 288 -23 197 27  T 106 176  Q 106 232 138 273  T 227 315  Q 260 315
                282 295  T 305 242  Q 305 195 270 160  T 188 125  H 178  Z "/></g><line transform="translate(0.00000, -717.49945)"
                x1="0" x2="468.78000" y1="0" y2="0" style="stroke-width:40.00000"/></g></g><path transform="matrix(0.48828, 0.00000,
                0.00000, -0.48828, 1579.88855, 0.00000)" d="M 154 272  Q 137 272 126 285  T 115 313  Q 115 330 126 342  T 154 354  H
                1440  Q 1455 354 1466 342  T 1477 313  Q 1477 298 1466 285  T 1440 272  H 154  Z M 154 670  Q 137 670 126 682  T 115
                711  Q 115 726 126 739  T 154 752  H 1440  Q 1455 752 1466 739  T 1477 711  Q 1477 694 1466 682  T 1440 670  H 154
                Z "/><path transform="matrix(0.48828, 0.00000, 0.00000, -0.48828, 2635.44702, 0.00000)" d="M 190 0  V 72  Q 446 72
                446 137  V 1212  Q 340 1161 178 1161  V 1233  Q 429 1233 557 1364  H 586  Q 593 1364 599 1358  T 606 1346  V 137  Q
                606 72 862 72  V 0  H 190  Z "/><path transform="matrix(0.48828, 0.00000, 0.00000, -0.48828, 3135.44702, 0.00000)"
                d="M 195 158  Q 243 88 324 54  T 498 20  Q 617 20 667 121  T 717 352  Q 717 410 706 468  T 671 576  T 602 656  T 496
                686  H 360  Q 342 686 342 705  V 723  Q 342 739 360 739  L 473 748  Q 545 748 592 802  T 662 933  T 684 1081  Q 684
                1179 638 1242  T 498 1305  Q 420 1305 349 1275  T 236 1186  Q 240 1187 243 1187  T 250 1188  Q 296 1188 327 1156  T
                358 1079  Q 358 1035 327 1003  T 250 971  Q 205 971 173 1003  T 141 1079  Q 141 1167 194 1232  T 330 1330  T 498 1364
                Q 560 1364 629 1345  T 754 1292  T 845 1204  T 881 1081  Q 881 995 842 922  T 737 796  T 590 717  Q 679 700 759 650  T
                887 522  T 936 354  Q 936 241 874 149  T 711 6  T 498 -45  Q 402 -45 305 -9  T 147 101  T 86 276  Q 86 327 120 361  T
                205 395  Q 238 395 265 379  T 308 336  T 324 276  Q 324 226 289 192  T 205 158  H 195  Z "/></g>
                </svg>
        TeV</a>
```

Figura 5.10: Imagen vectorial en el código fuente de un artículo en Google Scholar

Como es información innecesaria que complicaría la comparación de artículos, se decide eliminarlas. Estas imágenes están representadas por la etiqueta “<svg>”, por lo que se localizan con una expresión regular y se eliminan usando la función `sub` de la librería `re`.

Por otro lado, surge un problema al tratar con el sistema de duplicados de Google Scholar, ya que al asignar un artículo como duplicado no se puede usar la misma expresión regular que se estaba usando para obtener las citas porque la estructura del código fuente de la página cambia.

En la Figura 5.11, entre rectángulos azules, se puede ver este problema, donde el primer artículo es el repetido. Esto se soluciona añadiendo otra condición a la expresión regular, separadas por la expresión OR("|").

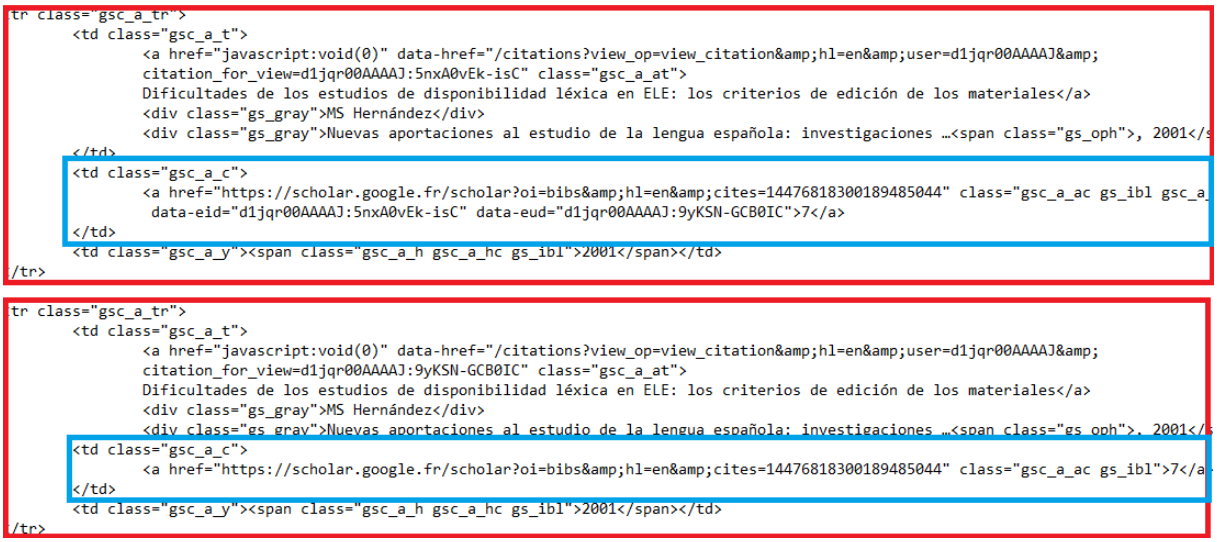


Figura 5.11: Ejemplo de artículos repetidos en Google Scholar

Otro inconveniente común que se puede encontrar al extraer información del código fuente son las HTML *entities*, las cuales son cadenas de texto que comienzan por el símbolo "&" y finalizan con un ";". Un ejemplo de HTML *entities* es "ũ" que traducido seria "ũ". Estas entidades se utilizan para mostrar caracteres especiales o difíciles de escribir en un teclado estándar, que de otro modo se mostrarían como código HTML. Una solución para evitar que caracteres especiales se extraigan como HTML *entities*, es la utilización de la librería `html` de Python junto con su función `unescape`, que permite traducir estas entidades a sus correspondientes caracteres.

Como último problema, nos encontramos que Google Scholar puede dejar etiquetas HTML sin eliminar en los datos de los artículos, tal y como se puede observar en la Figura 5.12. Entre estas etiquetas HTML se encuentran aquellas que se utilizan para enmarcar caracteres como superíndices o subíndices y también para mostrar el texto en cursiva o negrita. Para eliminarlas se ha usado la función `replace` para reemplazar estas etiquetas por un espacio en blanco, consiguiendo que los datos estén lo más limpios posibles.

Figura 5.12: Ejemplo del problema con etiquetas HTML en Google Scholar

5.2 Creación del módulo que relaciona artículos

Esta parte busca eliminar los artículos repetidos de cada autor, los cuales se encuentran en el archivo obtenido al finalizar el paso anterior **paper-cites-GS.csv** y agrupar los artículos restantes en uno solo. También se realizará este proceso a los artículos contenidos en el archivo **paper-cites-CRIS.xlsx** pertenecientes al repositorio institucional accedaCRIS y suministrado por la propia ULPGC. De cada uno de estos procesos se obtendrán los archivos **filtered_paper-cites-GS.xlsx** y **filtered_paper-cites-CRIS. xlsx**.

Para finalizar, se relacionan los artículos de ambos archivos ya filtrados, con el fin de obtener artículos que aparecen tanto en Google Scholar como en accedaCRIS, generando un archivo llamado **record_linkage_paper_cites.xlsx**.

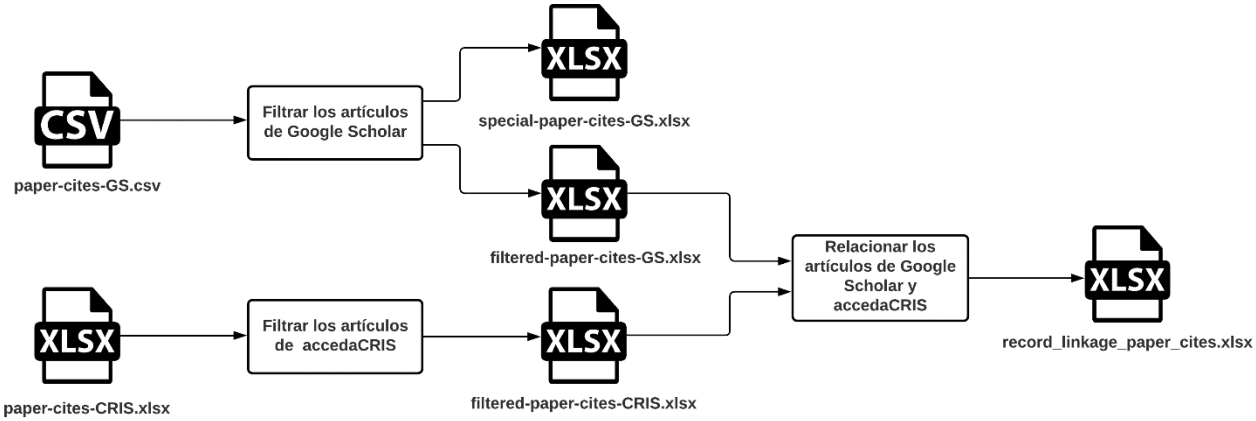


Figura 5.13: Diagrama de bloques de la Creación del módulo que relaciona artículos

(Fuente: Iconos realizados por Freepik - www.flaticon.com)

5.2.1 Filtrado de artículos de Google Scholar

En esta tarea se quitarán del archivo **paper-cites-GS.csv,** los artículos que se consideran repetidos y se agruparan en un solo artículo aquellos que se consideren iguales y pertenezcan a diferentes autores.

Para la realización de este proceso se usará la librería `Python Record Linkage Toolkit` que nos permite comparar los campos de un único conjunto de datos para encontrar duplicados o comparar los campos de dos conjuntos para relacionarlos.

Esta tarea comienza decidiendo que campos del archivo **paper-cites-GS.csv** se podrían comparar para buscar los artículos repetidos. Como se mostró en la Tabla 5.1, el archivo contiene los siguientes campos:

- **GS-Code:** código con el que Google Scholar identifica a los investigadores.
- **Art-Code:** código con el que Google Scholar identifica a los artículos.
- **Article:** título del artículo.
- **Authors:** autores que han participado en dicho artículo.
- **Journal:** revista donde se ha publicado el artículo. Puede contener el número de la revista, el número de páginas y la fecha de publicación.
- **Cites:** número de veces que se ha citado el artículo.
- **Year:** año en el que se publicó el artículo.

A continuación, se detallarán las razones por las cuales se deciden los campos a comparar:

El campo con el código del investigador se podría usar para comparar ya que los artículos que marcamos como repetidos, son aquellos que un investigador tiene por duplicados, pero más adelante se explicará porque este campo no se tendrá en cuenta para la comparación.

Con respecto al código del artículo, hay que destacar que es único e incluso si dos investigadores distintos tienen el mismo artículo, estos tendrán diferentes códigos, por lo que, no se puede tener en cuenta para comparar, pero sí se van a utilizar como índices para identificar los artículos.

El título del artículo es un campo que sí se tendrá en cuenta, ya que, junto al código del autor y del artículo, son campos que siempre va a tener un artículo y nunca va a estar vacío.

En relación con el campo de autores, se presentan dos problemas por los cuales no se debería elegir para realizar la comparación. Por un lado, un artículo puede tener este campo vacío y, por otro, el campo autores va a estar incompleto, ya que como se mencionó en el paso anterior, no se pueden recoger todos los autores sin acceder a la página propia de cada artículo y esto puede conllevar al bloqueo de la IP.

En cuanto al número de citas, tampoco es un campo que se deba tener en cuenta, ya que no siempre coinciden las citas que tienen dos artículos iguales.

Y por último, los campos de la revista y del año se pueden utilizar para realizar la comparación, aunque puedan encontrarse vacíos. Si no se tuvieran en cuenta, el título sería el único campo que se podría utilizar para realizar la comparación y sólo con él no se podría considerar que dos artículos son iguales.

A raíz de lo anterior, los campos que se utilizan para comparar los artículos repetidos son: el título del artículo, su revista y el año.

Como a partir de aquí se trabajará bastante con varios conjuntos de datos, se le asignará un nombre a cada uno para intentar no repetir mucho la palabra y que sea más comprensible. Los nombres que se utilizarán son los siguientes:

- **original_df:** contiene los datos tal cual se encuentran en **paper-cites-GS.csv**.
- **copy_df:** es una copia del original.
- **pairs_df:** contiene las parejas de artículos y sus comparaciones por los campos indicados.
- **match_df:** contiene las parejas de artículos que coinciden con las condiciones elegidas y se consideran como repetidos.
- **special_match_df:** contiene las parejas de artículos que coinciden con las condiciones elegidas y se consideran como especiales.
- **unique_df:** contiene los artículos de original_df que se consideran únicos después de eliminar los duplicados y especiales.
- **special_df**: contiene los artículos de original_df que se consideran especiales.

Una vez elegidos los campos, se comienza a preparar los datos para compararlos, pero antes se crea una copia de `original_df` usando la función `copy` de Pandas, generando así el conjunto, `copy_df`.

El campo del año no es necesario retocarlo, ya que al ser números se pueden comparar sin que se produzca ningún inconveniente, pero tanto el título como la revista, al ser cadenas de texto, hay que intentar que tengan el mismo formato para aumentar la precisión de la comparación. Para realizar este proceso, se hace uso del módulo `clean` de la librería que convierte todos los caracteres de una cadena de texto en minúsculas, elimina caracteres acentuados y elimina símbolos. Esta función se aplica a los campos del título y la revista de `copy_df` para no alterar los campos de `original_df`. En la Figura 5.14 se muestra un ejemplo en el cual se le aplica dicho proceso a tres títulos de diferentes artículos.

```
0                    Iron complexation by phenolic ligands in seawater
1    Nutrición y salud públicamétodos, bases científicas y aplicaciones
2                              Trypanosomosis in goats: current status
Name: Article, dtype: object
--------------------------------------------------------------------------------
0                iron complexation by phenolic ligands in seawater
1    nutricin y salud pblicamtodos bases cientficas y aplicaciones
2                           trypanosomosis in goats current status
Name: Article, dtype: object
```

Figura 5.14: Uso de la función clean en los títulos de tres artículos

Una vez aplicado el proceso en el título y en la revista de los artículos, ya tenemos preparado `copy_df` para empezar la comparación.

Lo primero es preparar el módulo de indexado que nos proporciona la librería de la librería con el módulo `index` y que se encarga de elegir las parejas de artículos que se van a comparar, dependiendo del tipo de algoritmo de indexación que se utilice. Hay varios algoritmos para

indexar, como el de *blocking* que permite, por ejemplo, en nuestro caso, comparar aquellos artículos que tienen el mismo código de autor. Para la realización de este proyecto se usa el algoritmo *full* que realiza todas las comparaciones posibles, pero sin repetirlas, es decir, sí el primer artículo ya se ha comparado con el segundo, el segundo no se va a comparar con el primero. Con esto se le indica al indexador que use este algoritmo en `copy_df` y se obtiene un *Multindex* de Pandas que contiene las parejas de índices de los artículos que se van a comparar.

```
       GS-Code                    Art-Code  ... Cites  Year
0  PGpIv6oAAAAJ  PGpIv6oAAAAJ:uJ-U7cs_P_0C  ...     2  2019
1  99Wc27MAAAAJ  99Wc27MAAAAJ:fEOibwPWpKIC  ...     2  2019
2  Yf596g8AAAAJ  Yf596g8AAAAJ:jq04SsiGh3QC  ...     2  2019

[3 rows x 7 columns]
MultiIndex([(1, 0),
            (2, 0),
            (2, 1)],
           )
```

Figura 5.15: Parejas de índices que calcula el indexador con tres artículos

Habiendo obtenido todas las parejas de artículos, el siguiente paso es el de elegir como se van a comparar dichas parejas por título, revista y año. Para ello se utiliza el módulo `compare`, que se encarga de comparar las parejas de artículos utilizando diferentes funciones. En el título y revista se usa la función `string` que nos mide el grado de similitud entre dos cadenas de texto usando diferentes métricas [12]. De estas métricas se usarán dos:

- **Levenshtein:** compara el grado de similitud entre dos cadenas de texto A y B, calculando el mínimo número de caracteres que se tienen que sustituir, eliminar o añadir, para conseguir que A sea igual a B. Tiene una complejidad de O(m*n), siendo m y n las longitudes de A y B, por lo que no se suele usar para comparar cadenas de texto largas, ya que tarda demasiado en realizar la comparación.

$$lev(A,B) = \begin{cases} |A| & if\ |B| = 0 \\ |B| & if\ |A| = 0 \\ lev\big(tail(A), tail(B)\big) & if\ A[0] = B[0] \\ 1 + min\begin{cases} lev(tail(A), B) \\ lev\big(A, tail(B)\big) \\ lev\big(tail(A), tail(B)\big) \end{cases} & otherwise \end{cases}$$

|X| es la longitud de una cadena de texto **X**.

tail(X) de una cadena de texto **X**, es la cadena de texto **X** menos el primer carácter.

Ecuación 5.1: Ecuación de la distancia Levenshtein

- **Jaro-Winkler**: esta métrica se divide en dos:

    - **Jaro**: busca el número de caracteres que tienen en común dos cadenas de texto, s_1 y s_2**,** teniendo en cuenta una distancia y que el orden de estos caracteres sea coincidente entre sí. Esta distancia es la mitad de la longitud de la cadena de texto más larga y sí el primer carácter de la cadena de texto más pequeña no se encuentra en esta, no se toma como una coincidencia válida.

$$sim_j = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & otherwise \end{cases}$$

    |S$_x$| es la longitud de la cadena de texto **X**.

    m es el número de caracteres que se consideran coincidentes.

    t es el número de caracteres que coinciden, pero no en el mismo orden y dividido entre 2.

Ecuación 5.2: Ecuación de la similitud Jaro

    - **Winkler**: se emplea después de realizar el proceso Jaro en los casos donde la puntación obtenida sea baja, pero las cadenas de texto comparten entre ellas una subcadena.

$$sim_w = sim_j + l\rho(1 - sim_j)$$

    l es la longitud de la subcadena común que comparten las dos cadenas abarcando hasta 4 caracteres.

    p es un factor de escala constante que determina cuánto se debe ajustar la puntuación por tener subcadenas comunes

Ecuación 5.3: Ecuación de la similitud Winkler

En el título se utiliza la métrica de Levenshtein, ya que aunque sea bastante costosa en tiempo al estar comparando cadenas de texto de gran longitud, ha dado mejores resultados que otras métricas.

En cuanto a la revista, se aplica la métrica Jaro-Winkler antes que la Levenshtein, puesto que existen casos donde dos artículos iguales y que se publican en la misma revista, uno de ellos, puede tener el número de la revista y las páginas donde está publicado el artículo y, el otro no, por lo que Levenshtein daría una puntación más baja y puede que no llegue a considerarlos iguales.

La función `string` junto a estas dos métricas devolverán valores entre 0 y 1, entendiendo por ejemplo 0.5 como un 50% de similitud entre las cadenas de texto.

Para comparar el año, se utiliza la función `exact` que compara si los años son exactamente el mismo, devolviendo un 0 cuando no coinciden y un 1 sí coinciden.

Especificadas las comparaciones en los campos elegidos, se utiliza la función `compute` del módulo `compare` perteneciente a la librería, que a partir del *Multindex* de Pandas que contiene las parejas de artículos que obtuvo el indexador y a `copy_df`, devuelve `pairs_df` que contiene la comparación de los artículos por los campos seleccionados tal y como se muestra en la Figura 5.16. A partir de ahora y como se indicó anteriormente, se utilizaran como índices los códigos de los artículos.

```
                            GS-Code                                                Article  ... Cites  Year
Art-Code                                                                                    ...
PGpIv6oAAAAJ:uJ-U7cs_P_0C  PGpIv6oAAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019
99Wc27MAAAAJ:fEOibwPWpKIC  99Wc27MAAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019
Yf596g8AAAAJ:jq04SsiGh3QC  Yf596g8AAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019

[3 rows x 6 columns]
MultiIndex([('99Wc27MAAAAJ:fEOibwPWpKIC', 'PGpIv6oAAAAJ:uJ-U7cs_P_0C'),
            ('Yf596g8AAAAJ:jq04SsiGh3QC', 'PGpIv6oAAAAJ:uJ-U7cs_P_0C'),
            ('Yf596g8AAAAJ:jq04SsiGh3QC', '99Wc27MAAAAJ:fEOibwPWpKIC')],
           names=['Art-Code_1', 'Art-Code_2'])
                                                  Article  Journal  Year
Art-Code_1                 Art-Code_2
99Wc27MAAAAJ:fEOibwPWpKIC PGpIv6oAAAAJ:uJ-U7cs_P_0C     1.0      1.0     1
Yf596g8AAAAJ:jq04SsiGh3QC PGpIv6oAAAAJ:uJ-U7cs_P_0C     1.0      1.0     1
                          99Wc27MAAAAJ:fEOibwPWpKIC     1.0      1.0     1
```

Figura 5.16: Comparación de tres artículos iguales

A partir de `pairs_df` que contiene las comparaciones, se clasifican los artículos en dos tipos: los repetidos y los especiales.

Se considera que dos artículos están repetidos, si los valores de las comparaciones cumplen que el título tenga una similitud igual o mayor al 90%, que la similitud de la revista sea mayor o igual al 80% y en el año al 100%.

Para considerarlos especiales, la coincidencia en el título tiene que ser igual o mayor al 90%, pero la revista debe de ser menor al 80% o el año ser 0%.

Estos porcentajes se fueron ajustando, basándose en la observación de los resultados obtenidos en las pruebas realizadas. Se comenzó estas pruebas con unos pocos artículos de manera que fue sencillo comprobar que se clasificaran como repetidos y especiales de forma correcta. Posteriormente se utilizaron 100 artículos y para comprobar que se clasificaban correctamente se generó un archivo con los artículos filtrados. A continuación, se abría este archivo con Excel y utilizando la opción de ordenar los artículos por título se comprobaba a mano que no hubiera ningún artículo repetido. En caso de que quedaran algunos artículos sin clasificar se tenían en cuenta para cambiar los porcentajes y se repetía el proceso. Este procedimiento se reprodujo también con 1300 artículos. Por último, se aplicó este proceso con todos los artículos, pero por la cantidad de artículos es posible que aún queden artículos que se tengan que eliminar.

La Figura 5.17 muestra cómo funciona el proceso de clasificación usando cuatro artículos iguales, pero uno de ellos tiene el año distinto. Lo que se encuentra dentro del rectángulo rojo es

match_df que contiene las parejas de artículos que clasificamos como repetidas por cumplir las condiciones antes mencionadas, y lo que se encuentra dentro del rectángulo azul es special_match_df que contiene las parejas que se consideran especiales.

Art-Code	GS-Code	Article	...	Cites	Year
d1jqr00AAAAJ:h1v_Rpz5TQ3w	d1jqr00AAAAJ	iron complexation by phenolic ligands in seawater	...	1	2018
PGpIv6oAAAAJ:uJ-U7cs_P_0C	PGpIv6oAAAAJ	iron complexation by phenolic ligands in seawater	...	2	2019
99Wc27MAAAAJ:fEOibwPWpKIC	99Wc27MAAAAJ	iron complexation by phenolic ligands in seawater	...	2	2019
Yf596g8AAAAJ:jq04SsiGh3QC	Yf596g8AAAAJ	iron complexation by phenolic ligands in seawater	...	2	2019

[4 rows x 6 columns]

Art-Code_1	Art-Code_2	Article	Journal	Year
99Wc27MAAAAJ:fEOibwPWpKIC	PGpIv6oAAAAJ:uJ-U7cs_P_0C	1.0	1.0	1
Yf596g8AAAAJ:jq04SsiGh3QC	PGpIv6oAAAAJ:uJ-U7cs_P_0C	1.0	1.0	1
	99Wc27MAAAAJ:fEOibwPWpKIC	1.0	1.0	1

Art-Code_1	Art-Code_2	Article	Journal	Year
PGpIv6oAAAAJ:uJ-U7cs_P_0C	d1jqr00AAAAJ:h1v_Rpz5TQ3w	1.0	1.0	0
99Wc27MAAAAJ:fEOibwPWpKIC	d1jqr00AAAAJ:h1v_Rpz5TQ3w	1.0	1.0	0
Yf596g8AAAAJ:jq04SsiGh3QC	d1jqr00AAAAJ:h1v_Rpz5TQ3w	1.0	1.0	0

Figura 5.17: Clasificación de 4 parejas de artículos en repetidos y especiales

Habiendo clasificado las parejas en dos conjuntos, se les aplica dos procesos distintos para decidir cuál de los dos artículos que forman la pareja, seleccionamos como repetido o especial.

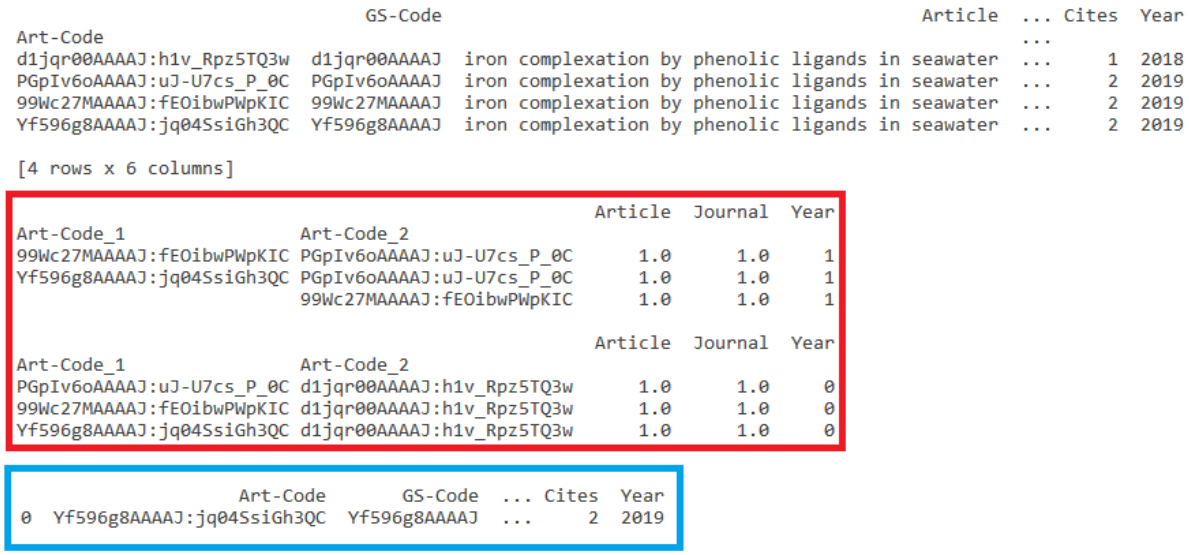
Comenzamos explicando el proceso de selección de artículos especiales, teniendo en cuenta que uno de los objetivos de este proceso es el de obtener artículos que tengan el mayor número de citas, ya que lo normal es que esos artículos sean verdaderos y no duplicados, al estar citados más veces. Considerando esto, se leen los índices de special_df, que son los códigos de los artículos, y a partir de estos, se usa la función loc de Pandas para acceder al campo de citas de original_df y compararlos. Todos los artículos que se consideren especiales se añadirán a una lista, por lo que, si uno de ellos tiene menos citas que el otro, se añade su código de artículo a la lista y cuando tienen el mismo número de citas, se tiene en cuenta el orden de aparición en original_df.

Continuamos, explicando el proceso para clasificar los artículos repetidos a partir de match_df**.** Como se ha mencionado queremos los artículos que tengan más citas, por lo que cundo se usó la función clean de Pandas, también se ordenó copy_df por el número de citas de menor a mayor, siendo siempre el último artículo el que tenga más citas. Por cómo calcula el indexador las parejas de artículos, siempre el último artículo va a compararse con el resto, pero ninguno con él, es decir, si se observa la Figura 5.17 se puede ver que el índice del artículo "Yf596g8AAAAJ:jq04SsiGh3QC" que aparece en match_df sólo aparece en el índice "Art-Code_1" por lo que si se cogen los códigos de los artículos del índice "Art-Code_2" y se eliminan, únicamente quedaría este artículo como un artículo único.

Con esto dicho, se añaden en una lista los índices de la columna "Art-Code_2" de match_df identificándolos como repetidos y usando la función isin de Pandas se quitan de

`original_df` todos los artículos que se encuentren en la lista de repetidos y en la lista de especiales, quedándonos únicamente artículos que consideramos únicos y formando `unique_df`.

La Figura 5.18 sirve como ejemplo de este proceso, haciendo uso de otros artículos para que cumplan las condiciones mencionadas. Dentro del rectángulo rojo se encuentran `match_df` y `special_match_df` respectivamente, y lo que se encuentra dentro del rectángulo azul, es lo que queda de `original_df`, después de haber quitado los artículos clasificados como repetidos y especiales.



Art-Code	GS-Code	Article	...	Cites	Year
d1jqr00AAAAJ:h1v_Rpz5TQ3w	d1jqr00AAAAJ	iron complexation by phenolic ligands in seawater	...	1	2018
PGpIv6oAAAAJ:uJ-U7cs_P_0C	PGpIv6oAAAAJ	iron complexation by phenolic ligands in seawater	...	2	2019
99Wc27MAAAAJ:fEOibwPWpKIC	99Wc27MAAAAJ	iron complexation by phenolic ligands in seawater	...	2	2019
Yf596g8AAAAJ:jq04SsiGh3QC	Yf596g8AAAAJ	iron complexation by phenolic ligands in seawater	...	2	2019

[4 rows x 6 columns]

Art-Code_1	Art-Code_2	Article	Journal	Year
99Wc27MAAAAJ:fEOibwPWpKIC	PGpIv6oAAAAJ:uJ-U7cs_P_0C	1.0	1.0	1
Yf596g8AAAAJ:jq04SsiGh3QC	PGpIv6oAAAAJ:uJ-U7cs_P_0C	1.0	1.0	1
	99Wc27MAAAAJ:fEOibwPWpKIC	1.0	1.0	1

Art-Code_1	Art-Code_2	Article	Journal	Year
PGpIv6oAAAAJ:uJ-U7cs_P_0C	d1jqr00AAAAJ:h1v_Rpz5TQ3w	1.0	1.0	0
99Wc27MAAAAJ:fEOibwPWpKIC	d1jqr00AAAAJ:h1v_Rpz5TQ3w	1.0	1.0	0
Yf596g8AAAAJ:jq04SsiGh3QC	d1jqr00AAAAJ:h1v_Rpz5TQ3w	1.0	1.0	0

	Art-Code	GS-Code	...	Cites	Year
0	Yf596g8AAAAJ:jq04SsiGh3QC	Yf596g8AAAAJ	...	2	2019

Figura 5.18: Conjunto obtenido después de eliminar artículos duplicados y especiales

Ahora solo quedaría agrupar los artículos únicos con los artículos que se compararon con estos durante el proceso de clasificación de repetidos, es decir, juntar sus códigos de autor de manera que todos los autores que han participado en la redacción del artículo quedan asociados.

Se ha desarrollado un método `group_author_codes` que va leyendo los artículos únicos de `unique_df` y a partir del código del artículo, busca en `match_df` aquellas parejas en las que este código se encuentre en el índice "Art-Code_1". Posteriormente, se cogen los códigos de autor de estos artículos y se añaden al campo código de autor del artículo único, verificando previamente si dicho código de autor es el mismo que el del articulo único para evitar que se repitan los códigos. Se puede ver el resultado de este proceso dentro del rectángulo azul de la Figura 5.19.

Como se mencionó anteriormente, este es el motivo por el cual se decidió no utilizar el campo código de autores para las comparaciones, ya que esta opción resulto ser más cómoda que añadir otro campo a la comparación y trabajar con él.

```
                              GS-Code                                                Article  ... Cites  Year
Art-Code                                                                                      ...
d1jqr00AAAAJ:h1v_Rpz5TQ3w  d1jqr00AAAAJ  iron complexation by phenolic ligands in seawater  ...      1  2018
PGpIv6oAAAAJ:uJ-U7cs_P_0C  PGpIv6oAAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019
99Wc27MAAAAJ:fEOibwPWpKIC  99Wc27MAAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019
Yf596g8AAAAJ:jq04SsiGh3QC  Yf596g8AAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019

[4 rows x 6 columns]
```

```
                                          Article  Journal  Year
Art-Code_1                Art-Code_2
99Wc27MAAAAJ:fEOibwPWpKIC PGpIv6oAAAAJ:uJ-U7cs_P_0C      1.0      1.0     1
Yf596g8AAAAJ:jq04SsiGh3QC PGpIv6oAAAAJ:uJ-U7cs_P_0C      1.0      1.0     1
                          99Wc27MAAAAJ:fEOibwPWpKIC      1.0      1.0     1
```

```
                   Art-Code       GS-Code  ... Cites  Year
0  Yf596g8AAAAJ:jq04SsiGh3QC  Yf596g8AAAAJ  ...     2  2019
```

```
                   Art-Code                                GS-Code  ... Cites  Year
0  Yf596g8AAAAJ:jq04SsiGh3QC  Yf596g8AAAAJ,PGpIv6oAAAAJ,99Wc27MAAAAJ  ...     2  2019
```

Figura 5.19: Conjunto final con los autores agrupados en un solo artículo

Para finalizar, se genera `special_df` que contendrá las parejas de artículos que consideramos repetidos. Estas parejas se obtienen usando las propiedades `index` y `values` de Pandas que nos permiten extraer los códigos de los artículos que se encuentran como índices en `special_match_df`. Se puede ver este resultado en la Figura 5.20.

```
                              GS-Code                                                Article  ... Cites  Year
Art-Code                                                                                      ...
d1jqr00AAAAJ:h1v_Rpz5TQ3w  d1jqr00AAAAJ  iron complexation by phenolic ligands in seawater  ...      1  2018
PGpIv6oAAAAJ:uJ-U7cs_P_0C  PGpIv6oAAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019
99Wc27MAAAAJ:fEOibwPWpKIC  99Wc27MAAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019
Yf596g8AAAAJ:jq04SsiGh3QC  Yf596g8AAAAJ  iron complexation by phenolic ligands in seawater  ...      2  2019

[4 rows x 6 columns]
```

```
                                          Article  Journal  Year
Art-Code_1                Art-Code_2
PGpIv6oAAAAJ:uJ-U7cs_P_0C d1jqr00AAAAJ:h1v_Rpz5TQ3w      1.0      1.0     0
99Wc27MAAAAJ:fEOibwPWpKIC d1jqr00AAAAJ:h1v_Rpz5TQ3w      1.0      1.0     0
Yf596g8AAAAJ:jq04SsiGh3QC d1jqr00AAAAJ:h1v_Rpz5TQ3w      1.0      1.0     0
```

```
                Art-Code_1                Art-Code_2
0  PGpIv6oAAAAJ:uJ-U7cs_P_0C  d1jqr00AAAAJ:h1v_Rpz5TQ3w
1  99Wc27MAAAAJ:fEOibwPWpKIC  d1jqr00AAAAJ:h1v_Rpz5TQ3w
2  Yf596g8AAAAJ:jq04SsiGh3QC  d1jqr00AAAAJ:h1v_Rpz5TQ3w
```

Figura 5.20: Conjunto con los artículos que se consideran especiales

En este proyecto no se trabajará con `special_df`, pero el objetivo es que, en un futuro mediante una interfaz, revisar a mano estos artículos clasificados como especiales y agruparlos con los únicos en caso de que se considere oportuno.

Esta tarea termina generando dos ficheros **filtered-paper-cites-GS** a partir de `unique_df` y **special-paper-cites-GS** a partir de `special_df`, finalizando el proceso de agrupación de artículos y, por lo tanto, se termina también la tarea respecto al archivo **paper-cites-GS.csv**.

5.2.2 Filtrado de artículos de accedaCRIS

Este proceso se lleva a cabo a partir del archivo **paper-cites-CRIS.xlsx**, que como se indicó, fue suministrado por la ULPGC y contiene todo el material académico de dicha institución. Este archivo contiene muchos más campos de los que se pueden extraer en Google Scholar. Entre los muchos campos que contiene el archivo **paper-cites-CRIS.xlsx** los más interesantes que se pueden usar para la comparación son:

- **Title:** título del artículo.
- **Handle:** código del artículo dado por la ULPGC.
- **ISSN:** código ISSN del artículo que nos permite identificar las publicaciones seriadas.
- **Year:** año en el que se ha publicado el artículo.
- **Journal_Title:** título de la revista donde se publicó el artículo.
- **DOI:** código DOI del artículo que nos permite identificar objetos digitales.
- **Author_List:** lista con los nombres de todos los autores que participaron en la publicación del artículo.
- **Author_Name:** el nombre del autor del artículo.
- **Author_Code:** código del autor del artículo.

Al tener más campos disponibles con los que realizar la comparación tenemos un abanico más amplio de opciones para comparar.

En esta ocasión se utilizan para la comparación el título, el ISSN, el DOI, el año, el título de la revista y la lista de autores. Salvo el título del articulo el resto de los campos podría estar vacío, pero más adelante se explicará el sistema que se utiliza para tener esto en cuenta.

Respecto al resto de campos tanto el nombre del autor como su código no son interesantes, y aunque en esta ocasión la gran mayoría de códigos de los artículos están relacionados entre sí, hay casos que no cumplen esta regla por lo que tampoco se utilizan para realizar la comparación. Hay que decir que, con estos datos, los códigos de los artículos no se pueden utilizar como índice para identificar a los artículos, ya que no son únicos, por lo que se usan las posiciones en las que se encuentran los artículos como índices.

A partir de aquí el resto del procedimiento se desarrolla prácticamente igual que en el procedimiento de filtrado de artículos de Google Scholar, pero con algunos cambios como, por ejemplo, que no se van a clasificar artículos como especiales, únicamente como duplicados. También se les asignará a los conjuntos de datos los mismos nombres que en el proceso anterior exceptuando los que tuvieran que ver con artículos especiales. Los nombres quedarían de la siguiente forma:

- **original_df:** contiene los datos tal cual se encuentran en **paper-cites-CRIS.xlsx**.
- **copy_df:** es una copia del original.

- **pairs_df:** contiene las parejas de artículos y sus comparaciones por los campos indicados.

- **match_df:** contiene las parejas de artículos que coinciden con las condiciones elegidas y se consideran como repetidos.

- **unique_df:** contiene los artículos de original_df que se consideran únicos después de eliminar los duplicados y especiales.

Después de elegir los campos, se crea una copia de `original_df` usando la función `copy` de Pandas generando así a `copy_df`. Como en esta ocasión no tenemos número de citas para priorizar la selección de los artículos, se ordena `copy_df` por el número de campos que no estén vacíos, por lo que siempre el último artículo será aquel que contenga más datos.

El título, la lista de autores y el título de la revista al ser cadenas de texto se tienen que dejar en el mismo formato para aumentar la precisión de la comparación por lo que se usa la función `clean` de Pandas en `copy_df`.

A continuación, se prepara el módulo de indexado utilizando el algoritmo *full* que aplicado a `copy_df`, obtiene un *Multindex* de Pandas que contiene las parejas de índices de los artículos que se van a comparar.

Seguidamente, se especifican con que algoritmos se realizaran las comparaciones de los campos de los artículos haciendo uso del módulo `compare`.

Para el título se utiliza la función `string` con el algoritmo Levensthein y para comparar la lista de autores y el título de la revista el algoritmo Jaro-Winkler. En el procedimiento anterior se explicó que la función `string` junto a estas métricas devuelven valores entre 0 y 1, pero en esta ocasión se le pasa también los porcentajes que queremos que tengan para considerarlos duplicados. Por ejemplo, si le ponemos que el título debe tener un 90% o más de similitud para decir que es repetido, cuando supera dicho umbral devuelve un 1 y sino lo cumple devuelve un 0. Con esto dicho, al título se le asigna un porcentaje del 90%, para la lista de autores un 80% y para el título de la revista un 85%.

Para el año, ISS y DOI se utiliza la función `exact` que compara si estos campos son exactamente el mismo, devolviendo un 0 cuando no coinciden y un 1 sí coinciden.

Una vez especificadas las comparaciones de los campos elegidos, se utiliza la función `compute` del módulo `compare`, que junto al *Multindex* de Pandas obtenido previamente y a `copy_df`, devuelve `pairs_df`, que contiene los resultados de la comparación de los artículos por los campos seleccionados.

A partir de `pairs_df`, se considera que dos artículos están repetidos si la suma de los resultados de todos los campos utilizados en la comparación es 3 o más y uno de esos campos tiene que ser el título. Las parejas de artículos que cumplan esta condición se encontraran en `match_df`.

Después se cogen de `match_df` los códigos de artículo que se encuentren en el índice "HANDLE_2" y se eliminan de `original_df`, usando la función `isin` de Pandas, los artículos que tengan estos códigos, obteniendo así `unique_df` que contiene los artículos únicos.

Por último, se utiliza el mismo método que en el proceso anterior `group_author_codes` para agrupar los artículos dentro de `unique_df`.

En la Figura 5.21 se muestra un ejemplo de todo este proceso, donde tenemos tres artículos iguales. Dentro del rectángulo rojo están `pairs_df` con las comparaciones entre los artículos y `match_df` con las comparaciones de artículos que consideramos duplicados, que en este caso son iguales. Dentro del rectángulo azul se encuentra el resultado final, que es un artículo único que agrupa a los dos autores.

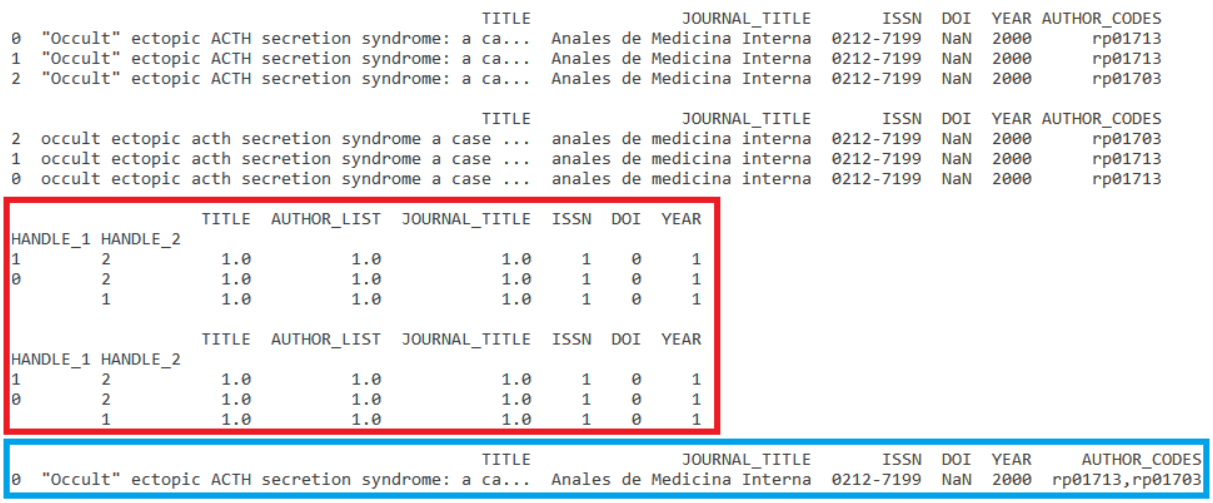


Figura 5.21: Ejemplo del funcionamiento del procedimiento con tres artículos

Con esto, se obtiene el fichero **filtered-paper-cites-CRIS.xlsx** a partir de `unique_df` y finaliza el proceso de eliminación y agrupación de artículos con el fichero **paper-cites-CRIS.csv**.

5.2.3 Relación de artículos de Google Scholar y accedaCRIS

Este último proceso de la creación del módulo que relaciona artículos consiste en buscar los artículos de Google Scholar que se encuentren también en accedaCRIS, utilizando los ficheros obtenidos en los dos procesos anteriores, con el objetivo de obtener una base de datos con artículos que se pueda asegurar que pertenecen a la ULPGC. Estos ficheros son **filtered-paper-cites-GS.xlsx** y **filtered-paper-cites-CRIS.xlsx**.

Como los campos que tienen en común los dos conjuntos son el título, la revista y el año, se usaran únicamente estos para realizar la comparación. Tanto el código del artículo en Google Scholar como en accedaCRIS se utilizarán como índices, ya que después de haber filtrado los artículos de accedaCRIS estos van a tener un código de articulo único.

De aquí en adelante, el proceso es exactamente igual a los dos anteriores y los nombres que se utilizarán para los conjuntos son los siguientes:

- **original_GS_df:** contiene los datos tal cual se encuentran en **filtered-paper-cites-GS.xlsx**.
- **original_CRIS_df:** contiene los datos tal cual se encuentran en **filtered-paper-cites-CRIS.xlsx**.
- **copy_GS_df:** es una copia del original_GS_df.
- **copy_CRIS_df:** es una copia del original_CRIS_df.
- **pairs_df:** contiene las parejas de artículos y sus comparaciones por los campos indicados.
- **match_df:** contiene las parejas de artículos que coinciden con las condiciones elegidas y se consideran como repetidos.
- **record_linkage_df:** contiene los artículos de Google Scholar que se encuentran también en accedaCRIS.

Se crean dos copias con la función `copy` de Pandas de `original_GS_df` y a `original_CRIS_df` y a estas copias se les aplica la función `clean` en los campos título y revista, para que tengan el mismo formato.

A continuación, se le indica al indexador que utilice la función full a partir de `copy_GS_df` y `copy_CRIS_df` obteniendo un *Multindex* de Pandas que contiene las parejas de índices de los artículos que se van a comparar.

Después se utiliza el módulo `compare` para realizar las comparaciones de estas parejas de índices. En el título se utiliza la función `string` con el algoritmo Levensthein y para comparar la revista el algoritmo Jaro-Winkler. La función `string` devolverá valores entre 0 y 1 dependiendo del grado de similitud entre los campos.

Para el año se utiliza la función `exact` que compara si los años son exactamente el mismo devolviendo un 0 cuando no coinciden y un 1 sí coinciden.

Habiendo especificado las comparaciones de los campos, se utiliza la función `compute` del módulo `compare`, que junto al *Multindex* de Pandas, `copy_GS_df` y `copy_CRIS_df`, devuelve `pairs_df` que contiene las parejas de artículos comparados por los campos utilizados con sus respectivas puntaciones.

A partir de `pairs_df` se consideran duplicados aquellos artículos cuyas puntaciones cumplan que el título sea igual o mayor al 90%, la revista sea mayor o igual al 80% y el año igual 100%, almacenando en `match_df` las parejas de artículos que cumplen estas condiciones.

Como lo que se busca son los artículos de Google Scholar que se encuentren también en accedaCRIS y ya en `match_df` están los que coinciden, se cogen los códigos de los artículos que se encuentran en el índice "Article" y usando la función `isin` de Pandas, nos quedamos con los artículos que se encuentran en `original_GS_df` que coincidan con estos códigos.

En la Figura 5.22 se puede ver todo este proceso, en el que se ha utilizado un conjunto de datos con tres artículos pertenecientes a Google Scholar y otro conjunto con dos artículos pertenecientes a accedaCRIS. Dentro del rectángulo rojo están `pairs_df` y `match_df`, y en el rectángulo azul se encuentra `record_linkage_df` que contiene los artículos que se encuentran en Google Scholar y en accedaCRIS.

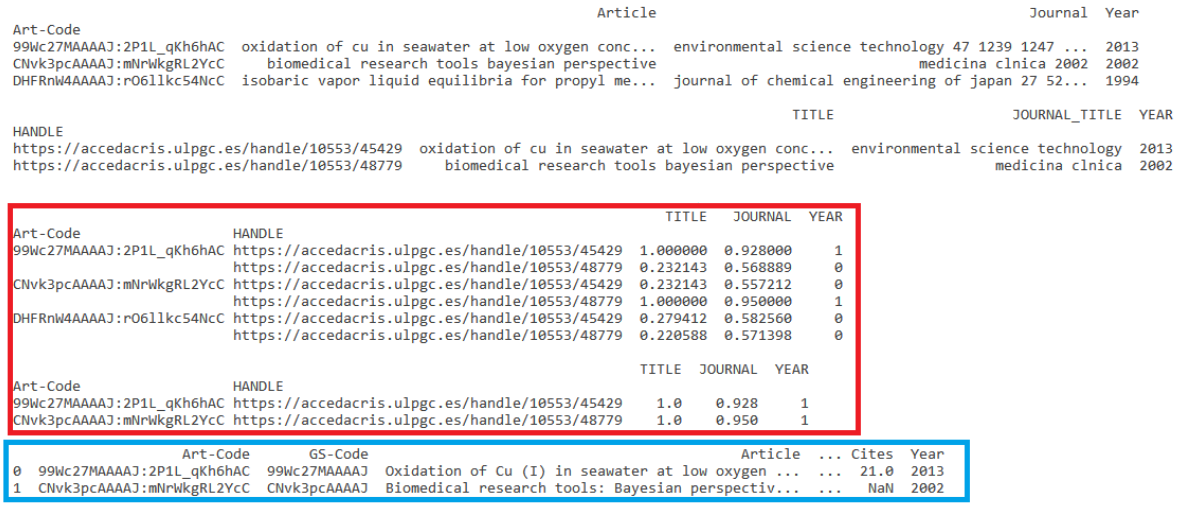


Figura 5.22: Relación de artículos correspondientes a Google Scholar y accedaCRIS

Con esto finaliza la creación del módulo que relaciona artículos, obteniendo el archivo **record_linkage_paper_cites.xlsx** y dejando todo lo necesario para continuar con el siguiente proceso.

5.3 Diseño de la red neuronal

En este apartado se desarrollará una red neuronal que permita clasificar los artículos como pertenecientes a la ULPGC o no. Para entrenar y probar esta red, se utilizará el archivo **record_linkage_paper_cites.xlsx** del proceso anterior, y artículos de otras instituciones, generando el conjunto de entrenamiento y el conjunto de prueba.

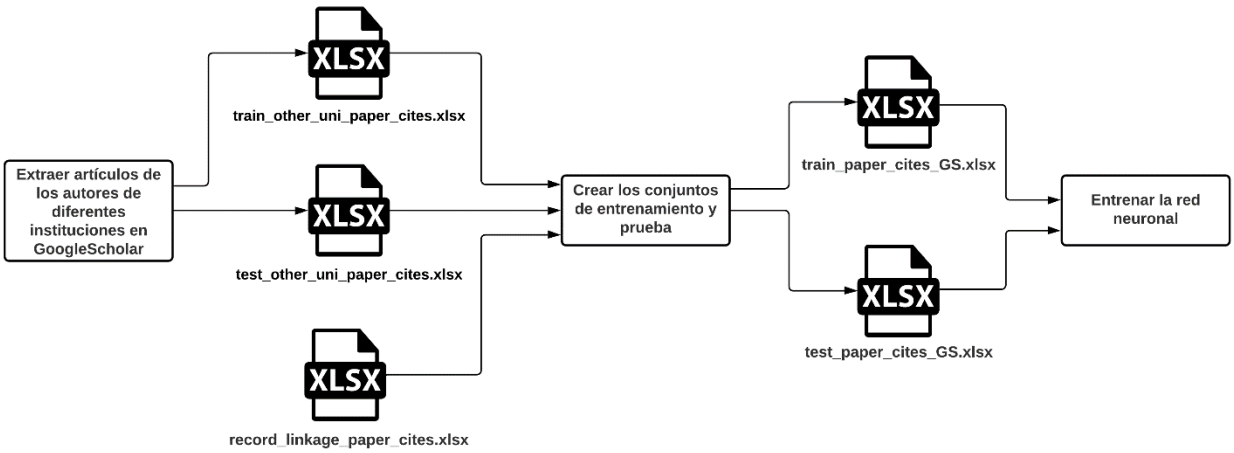


Figura 5.23: Diagrama de bloques del diseño de la red neuronal

(Fuente: Iconos realizados por Freepik - www.flaticon.com)

5.3.1 Generación de un conjunto de entrenamiento y prueba

A la hora de generar los conjuntos hay que tener en cuenta que el conjunto de entrenamiento y de pruebas deben tener artículos de distintas instituciones, ya que si ambos conjuntos tienen artículos pertenecientes a la misma institución, podría ocurrir que la red neuronal memorice los artículos y por lo tanto no se podría verificar que realmente está aprendiendo a clasificar cuales pertenecen a la ULPGC.

Teniendo esto en cuenta, el primer paso para crear estos dos conjuntos consiste en obtener los artículos de diferentes instituciones desde Google Scholar, utilizando el mismo método que se explicó en el apartado de creación de un módulo de descarga de datos. Este proceso se realiza extrayendo los artículos de los 10 primeros autores de las instituciones, ya que son suficientes artículos para entrenar a la red neuronal. De esto se obtienen dos archivos:

- **train_other_uni_paper_cites.xlsx:** contiene los artículos de autores que pertenecen a las universidades de Cambridge, Bremen y la universidad de California en Los Ángeles.
- **test_other_uni_paper_cites.xlsx** contiene artículos de autores pertenecientes a las universidades de Columbia y Princeton

Tras esto se añade al conjunto de datos obtenidos de **record_linkage_paper_cites.xlsx**, una columna llamada “Label”, que contendrá un 1 y servirá para identificar que estos pertenecen a la ULPGC.

Tabla 5.3: Conjunto record_linkage_df al añadirle la columna Label

Art-Code	GS-Code	Article	Authors	Journal	Cites	Year	Label
nyER5y0AAA	nyER5y0AAA	Primary prev	R Estruch, E F	New Englanc	5817	2013	1
nyER5y0AAA	nyER5y0AAA	Mediterrane	A Bach-Faig,	Public healtl	1238	2011	1
nyER5y0AAA	nyER5y0AAA	Food, youth	L Serra-Maje	Public healtl	949	2004	1
nyER5y0AAA	nyER5y0AAA	Scientific evi	L Serra-Maje	Nutrition rev	705	2006	1
nyER5y0AAA	nyER5y0AAA	Olive oil and	J López-Mira	Nutrition, m	583	2010	1

A continuación, a los otros conjuntos de datos **train_other_uni_paper_cites.xlsx** y **test_other_uni_paper_cites.xlsx** se les añade la misma columna, pero poniendo en "Label" un 0, indicando que estos no pertenecen a la ULPGC.

Para obtener el conjunto de entrenamiento se utilizan todos los artículos de **train_other_uni_paper_cites.xlsx**, ya que esto va a permitir evitar casos de *overfitting* durante el entrenamiento, por lo que habiendo añadido los "Label", comenzamos creando el conjunto de entrenamiento con la función `concat` de Pandas, que une **train_other_uni_paper_cites.xlsx** con **record_linkage_paper_cites.xlsx**.

En cuanto al conjunto de prueba, este debe tener el mismo número de artículos que pertenecen o no a al ULPGC, por lo que se extraen de **test_other_uni_paper_cites.xlsx** el mismo número de artículos que se encuentran **record_linkage_paper_cites.xlsx**, y luego se unen con la función `concat` de Pandas, generando el conjunto de prueba.

De esta manera generamos el archivo **train_paper_cites_GS.xlsx** a partir del conjunto de entrenamiento y **test_paper_cites_GS.xlsx** usando el conjunto de prueba, por lo que ya tenemos todo lo necesario para empezar el proceso de aprendizaje automático.

5.3.2 Entrenamiento de un clasificador binario

Para el entrenamiento de esta red neuronal se aplicará Word Embeddings [13], que es una técnica de procesamiento de lenguaje natural que permite buscar similitudes entre palabras vinculando a cada palabra un vector de números. El proceso de convertir las palabras en vectores se conoce como *tokenize* y es necesario realizarlo para que la capa *embedding* del modelo pueda analizar el texto.

Para todo este proceso se utilizarán los conjuntos de datos obtenidos en el paso anterior y que se encuentran en **train_paper_cites_GS.xlsx** y **test_paper_cites_GS.xlsx** Los campos que se van a utilizar para el aprendizaje son:

- **GS-Code:** el código de autor.
- **Articles:** título del artículo.
- **Authors:** lista de autores del artículo.
- **Journal:** la revista donde se publicó el artículo.
- **Year:** año de publicación del artículo.

Aunque la lista de autores esté incompleta, los nombres de los autores que se encuentran en su correspondiente campo sirven para el aprendizaje de la red neuronal.

Como se ha visto a lo largo de la explicación de este proyecto, algunos de los campos que se extraen de Google Scholar pueden estar vacíos por lo que antes de empezar con el proceso de *tokenize* se decidió rellenar estos campos para no perder información. Para el año se usa el método `mode` de Pandas que devuelve los valores que más aparecen en el campo indicado, por lo que se obtiene el año que más se repite en el conjunto y se completan los campos vacíos con él. En el caso de la lisa de autores y la revista, se optó por añadir simplemente una "a", ya que esto no afectara a los resultados obtenidos al quitar más adelante todas las cadenas de texto que tengan menos de tres caracteres.

El proceso de *tokenize* se realiza utilizando la clase *Tokenizer* de Keras. Esta clase crea un diccionario a partir de texto, donde cada palabra está asociada a un número. Entre los argumentos que se pueden utilizar a la hora de declararlo, se encuentran el número máximo de palabras que puede tener el diccionario, los filtros que se les aplica al texto, etc. En este caso no se le asigna un número máximo de palabras, pero se quitan los filtros por defecto, ya que al pasarle los códigos de autor, estos pueden contener símbolos y los filtros los quitarían perdiendo el código completo en el proceso. Como es importante que el texto tenga el mismo formato, se emplea una expresión regular que realiza la función de los filtros, pero aplicándose a todos los campos menos al de códigos de autor y, que además, como se explicó previamente se encargará de eliminar las cadenas de texto que tengan menos de tres caracteres.

Para crear el diccionario se usa el conjunto de entrenamiento que obtenemos de **train_paper_cites_GS.xlsx** y la función `fit_on_texts` del *Tokenizer*. Después de obtener el diccionario se aplica,a los conjuntos de entrenamiento y prueba, la función `texts_to_sequences` del *Tokenizer*, que convierte las palabras de los campos seleccionados en vectores con los valores que tienen asignadas en el diccionario. En la Figura 5.24 puede verse un ejemplo de todo este proceso aplicado de un artículo.

```
                                                        0
Art-Code                          nyER5y0AAAAJ:u5HHmVD_uO8C
GS-Code                                        nyER5y0AAAAJ
Article   Primary prevention of cardiovascular disease w...
Authors   R Estruch, E Ros, J Salas-Salvadó, MI Covas, D...
Journal   New England Journal of Medicine 368 (14), 1279...
Cites                                                5817.0
Year                                                 2013.0
label                                                     1

{'2013': 1, '<pad>': 2, 'nyer5y0aaaaj': 3, 'primary': 4, 'prevention': 5, 'cardiovascular': 6, 'disease': 7, 'with': 8,
mediterranean': 9, 'diet': 10, 'estruch': 11, 'ros': 12, 'salas': 13, 'salvadó': 14, 'covas': 15, 'corella': 16, 'arós':
17, 'new': 18, 'england': 19, 'journal': 20, 'medicine': 21, '368': 22, '1279': 23, '1290': 24}

  GS-Code               Article                     Authors                          Journal Year
0     [3]  [4, 5, 6, 7, 8, 9, 10]  [11, 12, 13, 14, 15, 16, 17]  [18, 19, 20, 21, 22, 23, 24, 1]  [1]
```

Figura 5.24: Ejemplo del uso de la clase Tokenizer de Keras

El objetivo de todo este proceso es unir todos los vectores de los campos en un nuevo campo, al que se le llamará “Tokenized_text”, y utilizar este para el proceso de aprendizaje, por lo que hay que asegurar que los vectores tengan el mismo tamaño. Para esto propósito, se le asigna a cada columna un tamaño fijo, de manera que se asegura que todos los datos de los artículos tengan el mismo tamaño, donde el tamaño del vector de la columna GS-Code será 1, el de la columna Article tendrá un tamaño de 25, el de la columna Authors será 15, el del campo Journal 15 y por último el del campo Year será 1. Los textos que no lleguen a este tamaño se les añadirá la palabra especial “<pad>” hasta completar el tamaño asignado, por lo que previamente se añade esta palabra especial al diccionario.

Para desarrollar el clasificador binario se utilizó un modelo sencillo, compuesto de una capa de entrada *embedding* y de una capa de salida *dense* con la función Sigmoide que devuelve valores entre 0 y 1. A la capa *embedding* se le ha dejado los parámetros que Keras fija por defecto, excepto los siguientes: el parámetro *input_dim* se le asigna el tamaño del diccionario obtenido anteriormente; al parámetro *output_dim* se le ha indicado que la dimensión de los vectores de salida de la capa sea 25 y al parámetro *input_length* se le asigna el tamaño de los vectores que se encuentran en el campo “Tokenized_text”, que al sumarlos el valor que le corresponde es 57.

Como se explicó anteriormente, el conjunto de entrenamiento está formado por más casos que no pertenecen a la ULPGC que casos que sí pertenecen. Para solucionar esto se utiliza el atributo *class_weight* durante el entrenamiento para indicarle al modelo que les preste más atención a los casos que sí pertenecen a la ULPGC consiguiendo que el conjunto de entrenamiento este más equilibrado.

A la hora de entrenar el modelo se dividió el conjunto de entrenamiento en dos, cogiendo un 20% de los artículos de este conjunto para usarlos en la validación del entrenamiento. Los parámetros utilizados en el compilador del modelo son el optimizador Adam, la función de perdida es *Binary Crossentropy* y la métrica empleada es *Accuracy*.

La Tabla 5.5 muestra la media de los resultados obtenidos después de ejecutar el modelo cinco veces con una única capa densa. Hay que remarcar que todas las ejecuciones fueron realizadas en 20 épocas y el tiempo promedio de cada una era de 50 segundos.

Tabla 5.4: Resultados del entrenamiento del modelo con una capa densa

Layer 1		Learning rate	Training Acuraccy	Validation Accuracy	Testing Accuracy
Neurons	Dropout				
64	0,2	0.01	0,9999	0,9991	0,9117
64	0,4	0.01	0,9999	0,9994	0,9064
64	0,6	0.01	**0,9998**	**0,9992**	**0,9418**
64	0,2	0.001	0,9999	0,9996	0,8183
64	0,4	0.001	0,9999	0,9997	0,8048
64	0,6	0.001	0,9999	0,9996	0,8406
64	0,2	0.0001	0,9998	0,9990	0,9286
64	0,4	0.0001	0,9998	0,9992	0,8218
64	0,6	0.0001	0,9999	0,9996	0,7303
16	0,2	0.01	0,9997	0,9992	0,8413
16	0,4	0.01	0,9997	0,9992	0,8461
16	0,6	0.01	**0,9996**	**0,9993**	**0,8745**
16	0,2	0.001	0,9998	0,9995	0,8446
16	0,4	0.001	0,9998	0,9995	0,8007
16	0,6	0.001	0,9998	0,9997	0,8138
16	0,2	0.0001	0,9998	0,9994	0,8380
16	0,4	0.0001	0,9996	0,9991	0,8033
16	0,6	0.0001	0,9996	0,9990	0,7580

Por los resultados de la precisión con los conjuntos de entrenamiento y validación, da la sensación de que se está produciendo un sobreajuste y que el modelo está memorizando los artículos, pero, aunque es un comportamiento extraño, por los resultados de la precisión del conjunto de prueba y examinando sus matrices de confusión se puede decir que no hay ningún sobreajuste y que el modelo está clasificando los artículos correctamente. También puede observarse que la precisión del conjunto de prueba oscila entre el 70% y 90%, además, de que a mayor número de neuronas posea la capa densa, mejor clasifica los artículos. Por otro lado, una ratio de aprendizaje de 0.01 da mejores resultados que otra de 0.0001.

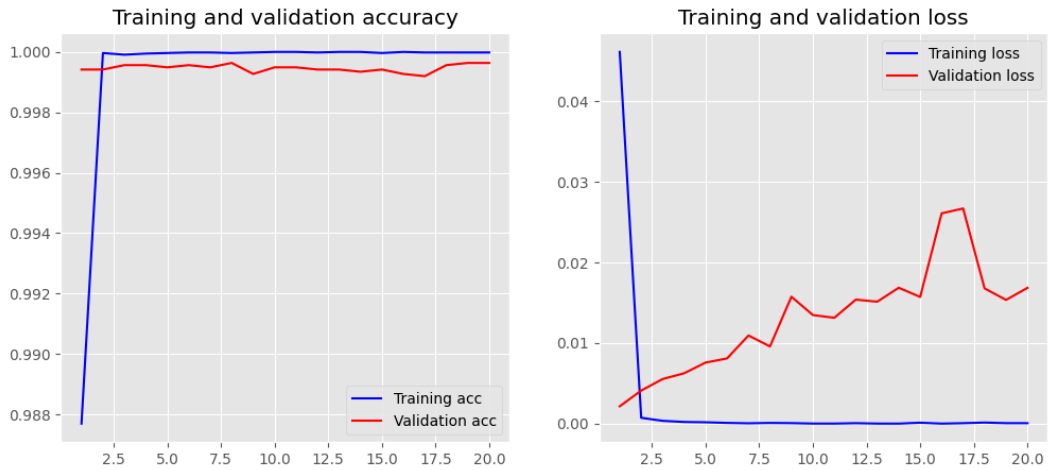


Figura 5.25: Gráfica de la precisión y la perdida de una de las ejecuciones con una capa densa

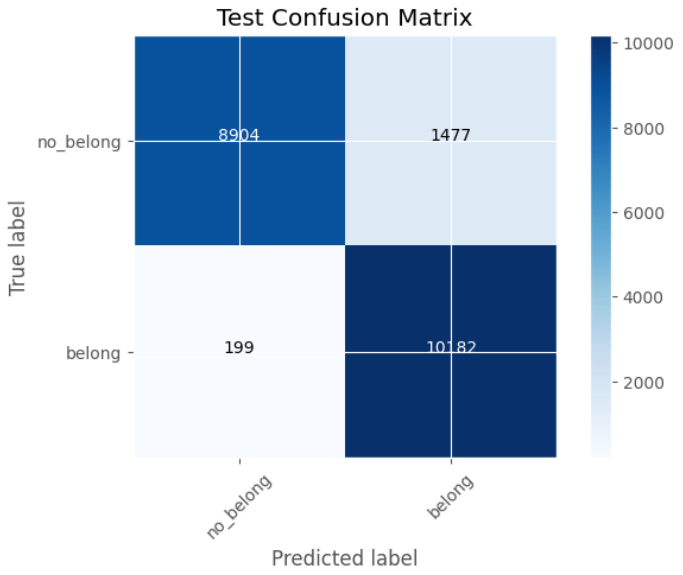


Figura 5.26: Matriz de confusión de una de las ejecuciones con una capa densa

Con el objetivo de comprobar si los resultados cambiaban al añadirle otra capa densa al modelo, se ejecutó el modelo cinco veces con dos capas densas y la media de los resultados obtenidos queda registrada en la Tabla 5.6. Todas las ejecuciones fueron realizadas en 20 épocas y el tiempo promedio de cada una era de 60 segundos.

Tabla 5.5: Resultados del entrenamiento del modelo con dos capas densas

Layer 1		Layer 2		Learning rate	Training Acc.	Validation Acc.	Testing Acc.
Neurons	Dropout	Neurons	Dropout				
64	0,2	**32**	0,2	0.01	**0,9998**	**0,9994**	**0,9223**
64	0,4	**32**	0,4	0.01	0,9997	0,9989	0,9038
64	0,6	**32**	0,6	0.01	0,9995	0,9992	0,8862
64	0,2	**32**	0,2	0.001	0,9997	0,9989	0,9139
64	0,4	**32**	0,4	0.001	0,9997	0,9991	0,9121
64	0,6	**32**	0,6	0.001	0,9996	0,9990	0,8674
64	0,2	**32**	0,2	0.0001	0,9997	0,9993	0,8519
64	0,4	**32**	0,4	0.0001	0,9997	0,9993	0,8285
64	0,6	**32**	0,6	0.0001	0,9997	0,9993	0,8273
32	0,2	**16**	0,2	0.01	**0,9998**	**0,9993**	**0,9022**
32	0,4	**16**	0,4	0.01	0,9997	0,9993	0,8919
32	0,6	**16**	0,6	0.01	0,9997	0,9990	0,8907
32	0,2	**16**	0,2	0.001	0,9998	0,9991	0,8912
32	0,4	**16**	0,4	0.001	0,9997	0,9992	0,8989
32	0,6	**16**	0,6	0.001	0,9997	0,9992	0,8462
32	0,2	**16**	0,2	0.0001	0,9998	0,9993	0,8081
32	0,4	**16**	0,4	0.0001	0,9997	0,9993	0,7929
32	0,6	**16**	0,6	0.0001	0,9999	0,9994	0,7902

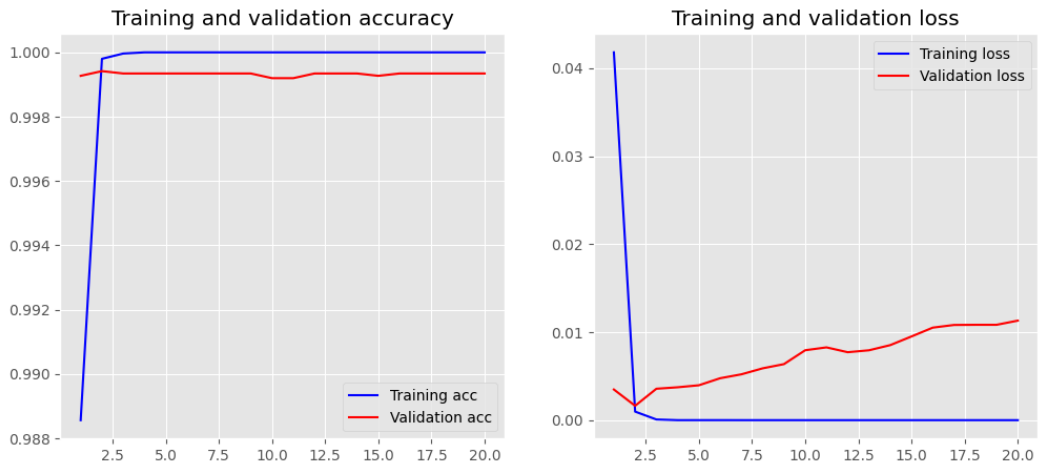


Figura 5.27: Gráfica de la precisión y la perdida de una de las ejecuciones con dos capas densas

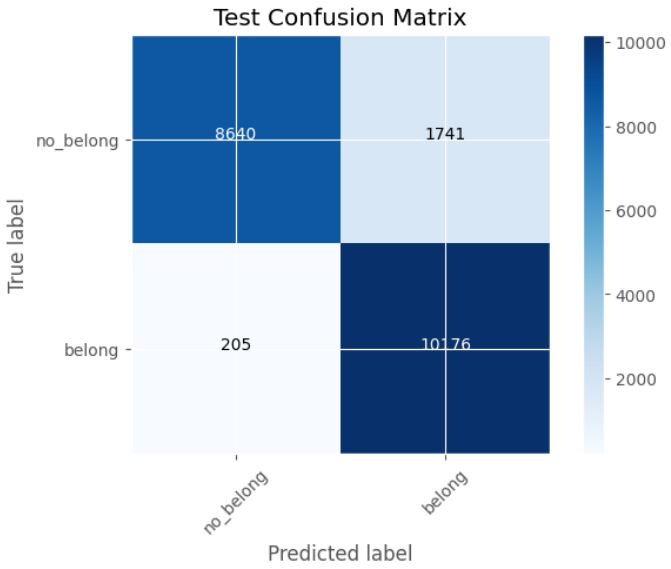


Figura 5.28: Matriz de confusión de una de las ejecuciones con una capa densa

Como se identificó con los anteriores resultados, a mayor número de neuronas en las capas densas y con una ratio de aprendizaje de 0.01 en el optimizador, los resultados mejoran, pero si se comparan con los obtenidos en la Tabla 5.1 no hay una mejoría significativa, por lo que añadir otra capa mas no ha ayudado a obtener mejores resultados.

Ante los resultados mostrados en ambas tablas se puede concluir que un modelo sencillo es capaz de clasificar los artículos como pertenecientes a la ULPGC correctamente. Aunque por falta de tiempo no se pudieron utilizar, la capas LSTM [14] podrían conseguir resultados más estables, ya que a estas se les da bien aprender secuencias al disponer de "memoria" mayor. También es posible utilizar *Word Embeddings* pre-entrenados [15] como, por ejemplo, Word2Vec o GloVe, con los que se puede obtener resultados más satisfactorios, al recoger un mayor número de palabras que el diccionario que se creó para este proyecto y además reducir el tiempo del proceso de entrenamiento.

Capítulo 6

Conclusiones y trabajos futuros

6.1 Conclusiones

Este proyecto ha sido un reto para el autor de este trabajo, ya que partía prácticamente de cero en los conocimientos necesarios para el desarrollo de este. Aprender a aplicar técnicas de *web scraping* o usar diferentes algoritmos para comparar cadenas de caracteres, fueron tareas complejas de aprender en un principio. Sin embargo, a medida que se iba superando la curva de aprendizaje inicial, se fueron implementado los diferentes objetivos planteados, consiguiendo avanzar satisfactoriamente en el desarrollo.

Tras haberle dedicado muchas horas a este proyecto, se considera que uno de los aspectos más importantes para conseguir el éxito del proyecto ha sido la correcta limpieza de los datos, tanto a la hora de extraerlos como antes de compararlos. Incluso con los problemas de que los datos estén incompletos se considera que se ha realizado un trabajo aceptable.

Para finalizar, se puede decir que este trabajo, no solo va a facilitar el trabajo a los departamentos de las bibliotecas universitarias y, por tanto, influir positivamente en las universidades, sino que también el alumno ha obtenido nuevos conocimientos y se ha asentado aquellos que se impartieron durante la carrera, pudiendo asegurar el éxito de un proyecto software.

6.2 Trabajos futuros

Aunque no se ha podido llevar a cabo en la realización del proyecto, como futura implementación se puede preparar un módulo que, de forma automática, importe los artículos que la red neuronal clasifique como pertenecientes a la institución y no se encuentren en la base de datos de esta.

También se podría retocar el módulo de eliminación de artículos repetidos y utilizarlo para quitar los artículos duplicados de las bases de datos de las instituciones, en caso de que no dispongan ya de un sistema similar.

Bibliografía

[1] P. Suber, Open Access, The MIT Press Essential Knowledge series, 2012.

[2] A. Robledano, «Qué es Python: Características, evolución y futuro,» 23 Septiembre 2019. [En línea]. Available: https://openwebinars.net/blog/que-es-python/. [Último acceso: 20 Mayo 2021].

[3] T. Rodríguez, «Las razones por las que muchos programadores están empezando a aprender Python,» 21 Abril 2020. [En línea]. Available: https://www.genbeta.com/desarrollo/estas-razones-que-programadores-estan-empezando-a-aprender-python-1. [Último acceso: 20 Mayo 2021].

[4] M. Abu Kausar, V. S. Dhaka y S. Kumar Singh, «Web crawler: a review,» *International Journal of Computer Applications,* vol. 63, nº 2, 2013.

[5] W. M. Hongkun Zhao y C. Yu, «Automatic extraction of dynamic record sections from search engine result pages,» 2006.

[6] T. Donohue, «DSpace-CRIS Home,» 15 Junio 2021. [En línea]. Available: https://wiki.lyrasis.org/display/DSPACECRIS/DSpace-CRIS+Home. [Último acceso: 3 Junio 2021].

[7] DuraSpace, «VIVO,» [En línea]. Available: https://duraspace.org/vivo/about/. [Último acceso: 3 Junio 2021].

[8] B. Alroe, «The PURE Institutional Repository: Ingestion, Storage, Preservation, Exhibition and Reporting,» *ELPUB,* pp. 455-456, 2007.

[9] E. Delgado López-Cózar, N. Robinson-García y D. Torres-Salinas, «The Google scholar experiment: How to index false papers and manipulate bibliometric indicators,» *Journal of the American Society for Information Science and Technology,* nº 65, pp. 446-454, 2014.

[10] J. P. Suárez Rivero, J. M. Doña Rodríguez, D. J. Greiner Sánchez, J. Sánchez Pérez y J. F. González Pérez, «ULPGC-Memorias de Investigación,» 2020. [En línea]. Available: https://www.ulpgc.es/vinvestigacion/memorias-investigacion. [Último acceso: 23 Junio 2021].

[11] D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato y F. Fdez-Riverola, «Web scraping technologies in an API world,» *Briefings in Bioinformatics*, vol. 15, p. 788–797, Sepiembre 2014.

[12] G. Wu, «String Similarity Metrics – Edit Distance,» 25 Noviembre 2020. [En línea]. Available: https://www.baeldung.com/cs/string-similarity-edit-distance. [Último acceso: 28 Mayo 2021].

[13] A. Bakarov, «A Survey of Word Embeddings Evaluation,» 21 Enero 2018.

[14] S. Hochreiter y J. Schmidhuber, «Long short-term memory,» *Neural computation,* vol. 9, nº 12, pp. 1735-1780, 1997.

[15] Y. Qi, D. Singh Sachan, M. Felix, S. Janani Padmanabhan y G. Neubig, «When and Why are Pre-trained Word Embeddings Useful?,» 2018.