

Estudio de un corpus de artículos científicos en economía usando técnicas de minería de datos



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Joaquín José Antón Orts

Tutor/es:

David Tomás Díaz

Pedro Albarrán Pérez



Universitat d'Alacant
Universidad de Alicante

Junio 2019

Agradecimientos

A David Tomás Díaz, por ayudarme en este ultimo paso del grado.

A Pedro Albarran Perez y María Angeles Carnero Fernandez, por aportarnos su experiencia y conocimientos sobre el dominio.

A mi familia, por estar ahí en los momentos mas duros.

A Jose Luís y Emilio, por su apoyo y ayuda en el desarrollo del proyecto

A Pavel, por sus consejos en el proyecto

A David, Jose (Castalla), Antonio, Luís, Eddie, Kalía, Jose (Petrer-Alicante), Núria, Javi, Fran y Cristian, por ser como una segunda familia y estar ahí siempre.

Índice de contenidos

1. Introducción y justificación del proyecto	7
2. Objetivos	9
3. Metodología	10
4. Estado de la cuestión	11
5. Obtención del corpus	14
5.1. ReDIF	14
5.1.1 Descripción del formato.....	14
5.1.2 Tipos de plantillas	17
5.2 Extracción y limpieza del corpus	20
5.2.1 Proceso de extracción	20
5.2.1 Limpieza del corpus.....	21
5.2.2 Problemas surgidos y soluciones aplicadas	22
5.3 Almacenamiento y consulta	24
5.3.1 APIs disponibles	27
5.4 Volcado de datos en Elasticsearch	33
5.5 Datos cuantitativos del corpus	34
6. Análisis	36
6.1 Kibana	37
6.1.1 Visualize	38
6.1.2 Dashboard.....	39
6.1.3 Dev tools	39
6.1.4 Management.....	40
6.2 Proceso de análisis	40
6.2.1 El camino del <i>working paper</i> a publicación	40
6.2.2 Análisis de software existente.....	51
7. Conclusiones	63
Referencias	65

1. Introducción y justificación del proyecto

Desde el siglo XIX, los avances y descubrimientos en las diferentes disciplinas científicas se difunden principal o casi exclusivamente mediante la publicación en revistas científicas (pese a que había pocas revistas desde el siglo XVII, ya que anteriormente se difundía principalmente mediante libros, en Matemáticas las primeras revistas son del primer cuarto del siglo XIX, en alemán, en Economía¹ las primeras son de mediados del siglo XIX, en alemán, y las principales son del último cuarto de siglo, ya en inglés). Es por ello que las revistas científicas han sido cruciales para entender los avances en cada una de las diferentes disciplinas y gracias a ello hoy en día se es posible tener acceso a todos estos avances. Cabe destacar que una gran parte del interés de los autores en las publicaciones científicas se debe a no solo a la difusión de avances y descubrimientos, sino que tanto las carreras laborales, promociones y prestigio tales como la financiación de investigadores, centros de investigación y universidades depende de la publicación de en revistas de prestigio y de las citas recibidas en otros artículos.

Enmarcados en el ámbito de la economía, hay una gran cantidad de artículos científicos publicados, muchos de los cuales se encuentran recopilados en RePEc (que es el repositorio del cual obtendremos los datos).

RePEc² (*Research Papers in Economics*), es un proyecto cuyo objetivo es recopilar la mayoría de publicaciones en el ámbito de la economía, así como sus versiones previas a dicha publicación e información acerca de los investigadores y sus centros de trabajo o investigación) para que de esta forma sea más accesible a una mayor cantidad de personas. Está compuesto por una gran base de datos bibliográfica que contiene información acerca de las diferentes contribuciones que están recopiladas.

A lo largo del desarrollo de este trabajo se realizará un análisis de artículos científicos de ámbito económico, para ello se dividirá el trabajo en dos partes, donde la primera de ellas

¹ La economía es la ciencia que estudia la elección entre usos alternativos de recursos escasos. Estas decisiones pueden hacerse por individuos (qué bienes comprar, en qué invertir, a qué dedicar el tiempo), empresas (qué bienes producir, con qué factores productivos: número de máquinas, trabajadores, ordenadores) o gobiernos.

² Página principal del proyecto: <http://repec.org/>

consistirá en obtener los datos que tiene almacenados el RePEc y en segundo lugar se realizará un análisis sobre dichos datos.

Con el objetivo de realizar un buen proceso de análisis, se ha contado con el apoyo y supervisión de varios profesores del Departamento de Fundamentos del Análisis Económico de la Universidad de Alicante.

Con la ayuda de dichos profesores, se han planteado diversas cuestiones de interés sobre el corpus de estudio, que durante el desarrollo de este trabajo nos hemos encargado de resolver, algunas de las cuestiones planteadas son las siguientes:

- ¿Cuanto tarda un *Working paper* en convertirse en *Article*?
- ¿Que aspectos económicos son los más tratados?
- ¿Que autores publican más?
- ¿Sobre qué temas habla cierto autor?
- Etc...

Para ello se realizarán una serie de procesos que nos proporcionarán información valiosa para los casos de análisis que se realizarán posteriormente. Los procesos, los cuales serán desarrollados más adelante, son los siguientes:

- Obtención de los ficheros desde el FTP de RePEc
- Implementación de un parseador de formato ReDIF³ a JSON⁴
- Preparación de un entorno para la ejecución de Elasticsearch⁵
- Volcado de estos documentos JSON en el servidor Elasticsearch
- Análisis de los documentos volcados
- Visualización de los resultados obtenidos en las técnicas de análisis.

³ ReDIF es un formato de estructuración de texto, cuya estructura básica está basada en el siguiente patrón: *Etiqueta: valor*. Se explicará con más detalle en el punto 5.1

⁴ JSON es un formato de texto cuyo principal objetivo es el intercambio de datos entre n individuos.

⁵ Elasticsearch es un software encargado del almacenamiento de datos de forma indexada. Se explicará con más detalle en el punto 5.3

2. Objetivos

Los principales objetivos de este proyecto son los siguientes:

- Aprendizaje propio en técnicas de minería de datos.
 - Se quiere obtener una mejor destreza en un campo tan amplio como es la minería de datos, de forma que en un futuro se puedan realizar ampliaciones más complejas de este trabajo e incluso aplicarlas a otros estudios en diferentes ámbitos científicos.
- Aprendizaje propio del uso de tecnologías de primer nivel para este campo.
 - Para complementar el objetivo anterior, es necesario dominar las tecnologías que son utilizadas en la gran mayoría de proyectos de este tipo (*Big Data* y *Analytics*), por lo que uno de los objetivos a cumplir es el aprendizaje y dominio de dichas tecnologías y herramientas.
- Realizar un análisis de datos en una ciencia tan importante como es la economía.
 - Uno de los dos objetivos más importantes a cumplir es la realización de un análisis sobre un corpus de gran tamaño e importancia como son los artículos científicos. Dicho análisis pasará por todas las fases (obtención, limpieza, almacenamiento, análisis y justificación) necesarias para el correcto desarrollo del mismo.
- Obtención de respuestas a las “preguntas” más interesantes realizadas a los datos.
 - Por último, el otro objetivo importante es la obtención de los resultados en los procesos de análisis y la justificación de los mismos, ya que esto es lo que nos permitirá decidir si el proyecto realizado ha cumplido las expectativas que teníamos antes de realizar el proyecto o no.

3. Metodología

El desarrollo de este trabajo conlleva unas fases que debemos seguir de forma ordenada. En nuestro caso han sido las siguientes:

1. Elección junto al tutor del proyecto de la temática
2. Cerrado el tema, se cuenta con la ayuda de profesores del departamento de Fundamentos del Análisis Económico para obtener la fuente de datos.
3. Una vez que tenemos claro el origen de los datos de estudio se detallan las tareas a seguir para volcarlas en una Base de datos propia.
4. A continuación se vuelve a contar con la ayuda de los profesores del departamento de Análisis Económico para plantear las cuestiones que se quieran responder a la consecución del trabajo.
5. Sabiendo las cuestiones a resolver en el desarrollo del proyecto, se dividen las tareas para realizar el proceso de análisis que lleve a cabo su resolución.
6. Por último, se justificarán los resultados obtenidos durante el proceso.

Para la consecución de estos objetivos, es necesario realizar una organización del trabajo a realizar, de forma que se establezca una jerarquía entre las distintas tareas a realizar. Es por ello que se ha empleado Trello para llevar el control de qué tareas se van realizando, la cual es gratuita y nos ofrece toda la funcionalidad que se necesita para este cometido.

En Trello definimos 3 columnas (TO DO, IN PROGRESS, DONE), de forma que organizamos las tareas para ver cuales hemos hecho y cuáles no.



Figura 1 - Tablero Trello para la gestión del proyecto

4. Estado de la cuestión

Desde los trabajos pioneros de J. D. de Solla Price y Eugene Garfield, cuyos trabajos establecieron la cienciometría moderna como disciplina científica (Price, 1978), el interés por un análisis riguroso de la producción científica ha crecido de manera paralela a la necesidad de obtener información sobre los artículos publicados, sus autores, centros de investigación y las revistas académicas. De hecho, ya en el año 1960 Eugene Garfield fundó el “Institute for Scientific Information” (ISI) para ofrecer servicios bibliométricos y de bases de datos bibliográficas: para cualquier investigador académico son bien conocidos su “Science Citation Index” (SCI), así como el “Social Sciences Citation Index” (SSCI) y el “Arts and Humanities Citation Index” (AHCI), todos ellos disponibles mediante el servicio de bases de datos “ISI - Web of Knowledge”. Por otro lado, la “American Economic Association” (AEA) ofrece un servicio de resúmenes de la literatura académica en Economía publicada en las principales revistas del campo de la Economía. El repositorio de RePEc (“Research Papers in Economics”) ofrece desde 1997 una base de datos descentralizada tanto de artículos publicados en revistas académicas como de “working papers”, “preprints” y componentes de software. Estas características lo distinguen y hacen único tanto respecto a los mencionados anteriormente como respecto al conocido repositorio ArXiv donde se envían (previa moderación) e-prints de trabajos científicos antes de su publicación definitiva de distintos campos, incluyendo Economía y otros como Estadística, Matemáticas, Física, Biología, etc. Zimmermann (2012) ofrece una descripción detallada de la recopilación de datos y el uso de estos para el cálculo de las clasificaciones dentro de RePEc.

El análisis cuantitativo de los artículos académicos publicados en Economía se remonta a Quandt (1976). Buena parte de la literatura sobre producción científica se ha centrado en analizar las citas recibidas por los artículos, como forma de medir su calidad, es decir, su impacto o mérito científico (Lindsey, 1989). Este enfoque se ha visto reforzado por el creciente uso de diversas medidas bibliométricas basada en citas, como el factor de impacto o el índice h, para ofrecer una valoración de la calidad de la publicación citada, su autor (y, por tanto, su centro de investigación) o el artículo que lo publica y, a partir de estas medidas, tomar decisiones relevantes de política científica, desde promociones de los investigadores hasta reparto de fondos. Hamersmesh (2018) ofrece una exhaustiva recopilación tanto sobre las fuentes y formas de índices basados en citas como sobre cómo se han empleado en Economía. Además de la construcción de índices individuales o agregados sobre producción científica, la literatura bibliométrica, en general, y algunos artículos en Economía en particular se han ido centrando en estudiar diversos aspectos de

los procesos de producción y de publicación de artículos científicos. Por un lado, algunos autores como Oswald (2007) y más recientemente el premio Nobel James Heckman (Heckman y Moktan, 2018) han presentado evidencia sobre los problemas de medir la calidad de un artículo y, por tanto, de un investigador a partir tanto de índices de las revistas (como el factor de impacto) como de otras medidas habituales basadas en citas. Otros artículos se han cuestionado también partes de proceso de publicación, ofreciendo evidencia sobre los posibles beneficios del “double-blind” en la evaluación por pares (Blank, 1991) o potenciales problemas de favoritismo en el proceso editorial (Laband y Pitette, 1994). En la mayoría de estos trabajos se han utilizado fuentes de información seleccionadas (unos pocos números de algunas revistas concretas), en lugar de usar repositorios sistemáticos como RePEc. Otra parte de la literatura se ha ocupado de caracterizar el proceso de citas, encontrando importantes diferencias entre distintas disciplinas científicas e incluso entre los sub-campos de la Economía (Wood, 2016). Usando información de EconLit, varios trabajos han encontrado algunos patrones interesantes sobre características de la producción científica en Economía. Ductor (2015) usa información de EconLit y, tras controlar por problemas de endogeneidad en la formación de redes, encuentra que la co-autoría beneficia la producción académica. Bramoullé y Ductor (2018) encuentran una fuerte y robusta relación negativa entre la extensión del título de un artículo y su calidad científica, en términos de las revistas en que se publica, citas recibidas (controlando por la calidad de la revista y de autores); además, la longitud del título se asocia negativamente con la novedad del artículo. Estos artículos se encuadran en la literatura que estudian cómo características menos profundas (i.e., relacionadas con calidad) de los artículos podrían afectar las frecuencias de citación. Estas características puede ser el resultado de prácticas académicas o de otras diferencias de artículos fácilmente identificables, como el orden alfabético de los autores (ver Einav y Yariv, 2006, y van Praag y van Praag, 2008) o su simplicidad (autores que comparten apellidos o apellidos comunes). Hengel (2019) también ha documentado diferencias de género en las citas recibidas por artículos publicados en las cinco revistas más importantes en Economía.

Otra rama de la literatura se ha dedicado a estudiar la evolución de distintas características de las publicaciones. Hammermesh (2013) documenta cambios en los patrones de coautoría, estructura de edad y metodología, resaltando la naturaleza cada vez más empírica de la investigación económica. Su muestra limitaba a 748 artículos de las tres principales revistas de economía (publicaciones de un año en cada década de 1960 a 2000). Angrist et al. (2017) utilizó técnicas de aprendizaje automático para ampliar este análisis a un conjunto mucho mayor de 135.000 artículos publicados en 80 revistas

académicas. Clasificar a mano cientos de miles de documentos en “teóricos” y “empíricos” sería prohibitivo, por lo que los autores utilizan “latent Dirichlet allocation” y “logistic ridge regression” para analizar la redacción de títulos y resúmenes y asignar cada documento a una categoría. Basado en un grupo más pequeño de cinco mil artículos clasificados por los asistentes de investigación, el algoritmo aprende qué palabras clave están asociadas con el trabajo empírico y el trabajo teórico y luego puede clasificar rápidamente miles de otros artículos que no fueron revisados directamente por los investigadores. Su trabajo confirma que la prevalencia del trabajo empírico ha ido en aumento en todos los sub-campos de la economía desde 1980. Los autores señalan que el giro empírico no es el resultado de que ciertos sub-campos más empíricos superen a otros más teóricos, sino de que cada sub-campo se vuelva más empírico.

Finalmente, varios artículos han analizado las características del proceso de publicación. Ellison (2002) documentó que el tiempo que un artículo de economía suele pasar en una revista entre su presentación y su publicación se ha más que duplicado en los últimos 30 años, de unos 8 a 16 meses. En línea con hallazgos empíricos anteriores de retrasos editoriales crecientes, tasas de aceptación decrecientes en las revistas y una tendencia hacia manuscritos más largos, Conley et al. (2011) estudian cómo el aumento de los retrasos en la publicación ha afectado al ciclo de vida de las publicaciones de los recientes doctores en economía. Construyendo un panel de 14.271 doctorados entre 1986 y 2000 en los departamentos de economía de EE.UU. y Canadá, encuentran evidencia de una disminución significativa de la productividad (medida en número de páginas equivalentes a páginas de “American Economics Review”). Card y DellaVigna (2013) presenta nuevas evidencias que confirman la percepción generalizada de que la publicación en las mejores revistas se ha hecho más difícil y mucho más lenta. Su trabajo muestra que el número de artículos publicados en las principales revistas ha disminuido, mientras que el número y la extensión de los envíos han aumentado.

5. Obtención del corpus

Para la obtención del corpus, se ha accedido al repositorio de RePEc⁶, el cual es una base de datos descentralizada que reúne artículos y otros elementos en el área económica. RePEc nos da acceso a más de un millón de publicaciones, la mayoría de las cuales son de acceso gratuito. La información recopilada en RePEc incluye datos sobre las citas recibidas por los artículos y los *Working papers*. Dichas citas son consideradas una medida de la “calidad” o importancia de las contribuciones científicas. El conjunto de esta información cruzada con la disponible en RePEc permite elaborar y publicar una serie de indicadores (como los conocidos como factor de impacto e índice-h así como sus variantes) sobre la producción científica de los autores y sus centros de trabajo. El uso de todos estos indicadores hace que se puedan elaborar clasificaciones de todo tipo, que van desde clasificaciones de investigadores y universidades hasta clasificaciones de las propias revistas, que son reconocidos por los investigadores académicos en Economía.

El acceso más sencillo y eficiente es mediante el acceso a un ftp que almacena toda la información sobre los artículos científicos, y mediante un comando rsync se han copiado todos los archivos residentes en dicho repositorio. Transcurrido el tiempo de descarga, se ha podido comprobar que obtenemos una estructura de directorios donde tenemos una gran cantidad de tipos de archivos diferentes (rdf, html, pdf, txt, etc...), debido a esto se ha realizado una selección de ficheros concreta, ya que por el momento únicamente nos interesan los archivos ReDIF (extensión “.rdf”) que son los que contienen los metadatos de los distintos documentos que hay recopilados (se explicará con más detalle en el punto 5.1).

5.1. ReDIF

5.1.1 Descripción del formato

El formato ReDIF (*Research Documents Information Format*) es un formato estructurado de datos, de forma que cada archivo con de este tipo, realmente es un fichero de texto en el cual se detallan una o más estructuras de datos, cada una de estas estructuras debe ser de un tipo concreto de plantilla, el cual está definido en la propiedad

⁶ Se puede obtener más información en el siguiente enlace:
https://es.wikipedia.org/wiki/Research_Papers_in_Economics

Template-Type. Dicha propiedad deberá tener el siguiente formato: “ReDIF-<tipo> <versión>”, donde <tipo> define el tipo de contribución que se aporta (article, paper, software, etc...) y <versión> nos indica la versión de ReDIF que se utiliza (normalmente 1.0).

Este campo nos sirve para saber que campos extraer en cada caso, ya que no podemos generalizar este comportamiento debido a que cada tipo contiene unos campos u otros (por ejemplo, un tipo de plantilla *person* no contiene un campo Abstract, por lo que al intentar extraerlo se producirá un error).

El formato de las propiedades es siempre el mismo de forma que esto nos ofrece mayor facilidad a la hora de extraer los datos de los ficheros, este formato debe seguir la siguiente estructura:

- *propiedad: valor*

Este formato nos ofrece una gran cantidad de datos interesantes para los procesos de análisis, ya que nos provee de casi todos los metadatos del artículo (Título, Autor/es, Resumen, Clasificación JEL, etc...) .

Respecto a este formato, hay que destacar también las propiedades llamadas *clusters*, las cuales nos informan de un conjunto de datos referentes a una entidad en particular.

Por ejemplo, si tenemos el siguiente grupo:

Author-Name: Joaquín

Author-Email: joaquin.anton95@gmail.com

Author-Phone: 612345678

Nos está indicando que toda esa información que nos está dando pertenece al autor de la contribución. Esto nos ofrece una gran facilidad a la hora de obtener los datos, ya que en caso de tener coautorías en los registros, podemos saber a qué o quién corresponde cada conjunto de datos.

Por ejemplo, en este caso tenemos una contribución que pertenece a dos autores:

Author-Name: Smith, Adam

Author-Email: Adam.Smith@classical.econ.org

Author-Name: Ricardo, David

Author-Email: Ricardo@classical.econ.org

Gracias al sistema de *clusters*, podemos saber que las 2 primeras propiedades pertenecen al autor "Adam Smith" y las 2 siguientes al autor "David Ricardo", por lo que podemos extraer los datos y almacenarlos con mayor facilidad.

También podemos anidar *clusters*, por lo que podemos obtener información de un *cluster* dentro de otro, la estructura es la siguiente: **<propiedad>-<subpropiedad>-<Dato>: <valor>**. Por ejemplo tenemos la propiedad *Author-Workplace-Name*, en la cual la propiedad principal nos indica que se está hablando del Autor (Author), la secundaria nos indica que es el lugar de trabajo (Workplace) y la tercera nos indica que el dato concreto es el Nombre, por lo que este *cluster* nos informa del nombre del lugar de trabajo del autor.

Por ejemplo, en un caso práctico como el siguiente:

Author-Name: Jakob Roland Munch

Author-Name: Michael Svarer

Author-Email: [msvarer@econ.au.dk](mailto:m svarer@econ.au.dk)

Author-WorkPlace-Name: Department of Economics, University of Aarhus, Denmark

Author-WorkPlace-Postal: 8000 Aarhus C, Denmark

Author-WorkPlace-Fax: +45 86 13 63 34

Al igual que en el caso anterior a este, tenemos 2 autores, por lo que no podríamos saber a que se refiere cada información, pero gracias a los *clusters* anidados podemos saber que ambos autores pertenecen a la misma institución y los datos de esta, de forma que nos permite obtener más de un dato sobre el *cluster* que estamos tratando. En este caso, además del nombre podemos saber la dirección de la institución así como su fax.

5.1.2 Tipos de plantillas

Tenemos 3 grandes grupos de plantillas en todo el formato ReDIF, los cuales son los siguientes:

- Colecciones
- Entidades físicas
- Recursos

Dentro de las colecciones tenemos los siguientes subtipos:

- *Archive*
- *Series*

Con el primero de ellos podemos obtener información de cómo se reflejan los datos bibliográficos de los artículos científicos que pertenecen a la colección. Además también nos ofrece información de la persona o personas que lo mantienen.

En cambio, el subtipo *Series* nos ofrece una información más detallada y útil para el usuario final, ya que además de proporcionarnos un nombre y una descripción de la colección, nos ofrece información acerca de la entidad que provee esta colección, de forma que el usuario que haga uso de estas colecciones sabe en todo momento de dónde viene dicha colección sin la necesidad de que esta información esté reflejada en todos y cada uno de los artículos.

A continuación vamos a mostrar un ejemplo del subtipo *Series*:

Template-Type: ReDIF-Series 1.0

Name: DRUID Working Papers

Provider-Name: DRUID, Copenhagen Business School,

Department of Industrial Economics and Strategy/Aalborg University, Department of Business Studies

Provider-Homepage: <https://www.druid.dk/>

Maintainer-Name: Keld Laursen

Maintainer-Email: kl.ivs@cbs.dk

Handle: RePEc:aal:abbswp

Como podemos ver, el ejemplo nos da información acerca de la institución a la que pertenece la colección, tales como el nombre de la misma, su página web y datos sobre la persona encargada de mantener dicha colección.

Respecto al grupo de Entidades físicas, tenemos los siguientes subtipos:

- *Person*
- *Institution*

El primero de ellos hace referencia, como su nombre indica, a personas físicas, de forma que podemos obtener la información básica de esta persona. De aquí se pueden extraer datos como el nombre, apellidos, página web, etc...Por ejemplo:

Author-Name: Thomas Krichel

Author-Name-First: Thomas

Author-Name-Last: Krichel

Author-Person: pkr1

En este caso, se nos ofrece los campos *Name*, *Name-First*, *Name-Last* y *Person*, el primero de ellos es el nombre completo de la persona, los 2 siguientes nos ofrecen por separado el nombre y apellidos, y el último es una referencia para identificarlo (similar al *Handle* de los documentos)

Respecto al tipo *Institution*, destacar que es similar al tipo anterior, con la diferencia de que aquí podemos obtener mucha menos información, ya que campos como teléfono, clasificación o lugar de trabajo entre otros no se pueden reflejar según la guía ReDIF. La información que podemos extraer sobre la institución correspondiente comprende hasta cuatro niveles de profundidad en la jerarquía de la institución, como por ejemplo:

Primary-Name: Université des Grands Espoirs

Primary-Location: Panava-les-Flots

Secondary-Name: Departement d'Économie

Secondary-Email: eco@uge.edu

Secondary-Homepage: <http://www.eco.uge.edu/>

Secondary-Phone: (+567)3466356

Este ejemplo nos ofrece información sobre la universidad y el departamento en concreto que ha realizado la publicación.

Por último, destacar las plantillas (*Templates*) más relevantes y en las que nos vamos a centrar en el desarrollo de este trabajo, las cuales son también las que más presencia tienen en el repositorio de RePEc, que son las del grupo de Recursos:

- *Paper*
- *Article*
- *Software*
- *Book*
- *Chapter*

De todos ellos podemos extraer casi la misma información, ya que en este tipo de contribuciones casi siempre tenemos campos comunes como son el Título, el nombre del Autor y el resumen de la contribución.

El primero de ellos (*Paper*) se refiere a artículos que no están publicados en ninguna revista, ni como un libro ni como parte de uno. En este caso la guía ReDIF nos dice que esta contribución puede contener campos como *Title*, *Abstract*, *Author-Name*, etc..

El siguiente (*Article*) nos ofrece prácticamente la misma información que el anterior, pero la diferencia entre ambos se basa en que esta contribución sí que ha sido extraída de alguna revista del campo de la economía. Es por ello que se realiza dicha separación entre ambas contribuciones.

Los 3 subtipos que quedan, son los que menos presencia tienen en el conjunto de este grupo de Recursos. No obstante, los tipos libro (*Book*) y capítulo (*Chapter*) siguen teniendo gran peso en el repositorio- Es por ello que los hemos incluido en nuestro proceso de análisis. Por último, ya que el trabajo se enmarca en el ámbito del desarrollo software, también se ha decidido incluir las aportaciones que sean de este tipo (*ReDIF-Software*).

De estos 3 subtipos de contribuciones, podemos extraer información bastante útil para la posterior fase de análisis, de modo que podemos obtener datos como el año de publicación o el estado de la misma tanto para el tipo *Book* como para el tipo *Chapter*, además de obtener información sobre el libro en el que está contenido este último tipo. Por supuesto, además de los campos anteriormente mencionados, también tenemos presencia de los campos básicos de casi todas las plantillas, tales como *Author-Name*, *Title* o *Abstract* entre otros.

En el caso del tipo *Software*, el campo a destacar es *Programming-Language*, el cual nos indica en qué lenguaje de programación se ha desarrollado el software que ha sido publicado en el RePEc, además de los campos básicos anteriormente nombrados.

5.2 Extracción y limpieza del corpus

5.2.1 Proceso de extracción

Una vez leídos los ficheros con extensión “.rdf” (más de 760.000), se ha extraído la información en formato ReDIF, el cual hemos explicado anteriormente. En cada uno de estos ficheros encontramos metadatos de uno o más artículos científicos. De cada uno de estos artículos conocemos la siguiente información:

- **Title:** Título del artículo
- **Author-Name:** Nombre del autor del artículo (si son varios autores, tendremos una línea por cada uno de ellos).
- **Author-Email:** Email del autor (al igual que el nombre, puede contener varios)
- **Author-WorkPlace-Name:** El lugar de trabajo del autor
- **Author-WorkPlace-Postal:** Dirección del lugar de trabajo del autor
- **Author-WorkPlace-Fax:** Fax de dicho lugar
- **Abstract:** Resumen breve del artículo.
- **Classification-JEL:** Clasificación del artículo según los códigos de la *American Economic Association*, autores de la clasificación JEL.
- **Keywords:** Palabras clave para identificar el artículo.
- **Length:** Número de páginas del mismo.
- **File-URL:** Ruta donde está almacenado el fichero (normalmente es el mismo FTP)
- **File-Format:** Formato del fichero (representado como tipo MIME)

Para la extracción de cada uno de los artículos, se han leído todas las líneas del fichero y se han procesado cada una de estas por separado. Para cada línea separamos por el carácter ‘:’ y comprobamos si el resultado obtenido son dos tokens, en caso de ser así nos encontramos con el caso más básico del procesado, ya que el primer token es la propiedad y el segundo token el valor de la misma. En caso de encontrar un único token (no hay separación por el carácter ‘:’) o más de dos tokens (hay más de un carácter ‘:’) hay que procesarlos de forma diferente, de manera que en el primer caso se añade el token que tenemos al final del valor de la última propiedad que hemos leído (esto suele darse en el procesamiento del campo “Abstract”, ya que puede estar compuesto por varias líneas del

fichero). En el segundo supuesto, establecemos el primer token como la propiedad y los siguientes tokens los concatenamos entre sí separados por el carácter ':' y los establecemos como valor de dicha propiedad (este caso suele producirse cuando tenemos que procesar propiedades como las URLs, ya que estas contienen el carácter ':' después del protocolo a utilizar: http, https, ftp, etc...).

5.2.1 Limpieza del corpus

Durante el proceso de extracción, se han ido realizando pequeñas pruebas de concepto, las cuales nos han permitido descubrir que uno de los principales problemas que teníamos en el corpus era la cantidad de nombres de propiedades erróneas que había en el corpus, ya que pese a que hay una guía de etiquetado⁷ con las pautas y nombres a seguir, muchos de los creadores no la siguen estrictamente.

Se ha realizado un proceso de limpieza con estas propiedades, de forma que gracias al mismo podemos tener un corpus más homogéneo, lo cual nos facilitará el proceso de análisis.

La limpieza del corpus se basa en aplicar 2 grandes comprobaciones a todas y cada una de las propiedades, las cuales son las siguientes:

- Primero comprobamos que la propiedad está dentro del listado de propiedades que hemos mencionado antes (más frecuentes).
- En caso de no encontrarla, se busca la propiedad en una lista de equivalencias creada previamente.

Para la primera comprobación, como ya ha sido mencionado, se ha realizado un primer recorrido por las propiedades y se han volcado en un fichero, el cual posteriormente ha sido revisado. Con esto se quería conseguir ver qué propiedades salen con más frecuencia y cuales son menos comunes y realizar una selección de las mismas. Esta selección nos permite establecer los criterios de volcado para la primera comprobación mencionada, del mismo modo que nos permite saber qué propiedades están mal escritas y de ese modo realizar una segunda lista con equivalencias de las propiedades erróneas a sus valores correctos. Por ejemplo:

⁷ Aquí podemos ver la guía a seguir: http://openlib.org/acmes/root/docu/redif_1.html

- Author-1-Name → Author-Name
- author-x-email → Author-Email

En estos ejemplos, como podemos ver no siguen la guía de etiquetado de ReDIF, por lo que en estos casos mapeamos los campos a valores correctos

Esta lista de equivalencias realizada, asienta las bases de la segunda comprobación, en la cual comprobamos que la propiedad mal escrita esté en dicha lista y en caso afirmativo se obtiene el valor mapeado correcto, el cual se utiliza para los procesos de volcado.

De esta forma conseguimos que todo lo que se vuelque en nuestra base de datos local sean valores controlados, por lo que no tendremos dificultades de este tipo en los procesos de análisis.

5.2.2 Problemas surgidos y soluciones aplicadas

5.2.2.1 *Encoding* de los ficheros

El principal problema encontrado aquí, ha sido el encoding de los ficheros, ya que no todos están en UTF-8, por lo que en el momento de la lectura de cada uno de los ficheros se ha tenido que detectar el encoding mediante el uso de una librería en python⁸, la cual nos proporciona el encoding del fichero con un valor de confianza del mismo, es decir nos indica en qué grado de seguridad es cierta dicha codificación. Realizando esta comprobación, se ha detectado que la gran mayoría de ficheros en los que obtenemos un porcentaje de confianza inferior al 50% producían fallo de lectura, ya que el encoding obtenido no es el real del fichero. En caso de producirse fallo de lectura hemos tenido que ignorar estos ficheros, ya que no podíamos asegurar al 100% que los datos extraídos fueran correctos (sin caracteres extraños). Esto se ha podido corroborar aplicando el comando de unix “file <fichero>” (el cual nos indica la información del fichero) ya que el resultado de este comando es diferente al obtenido mediante la librería de python.

El número total de ficheros que no estaban en UTF-8 es de 7973 ficheros, de los cuales un total de 566 han tenido que ser excluidos por no conseguir asegurar la codificación que estos tenían.

⁸ Se ha empleado la librería *chardet* en su última versión. <https://pypi.org/project/chardet/>

5.2.2.2 Contenido de los ficheros

Como ya hemos mencionado previamente, en el apartado 5.2.1, muchos usuarios de los que contribuyen en este repositorio, no cumplen las especificaciones de la guía a la hora de aportar su recurso a la colección. Esto conlleva que tengamos inconsistencias a la hora de realizar un análisis de estos recursos.

Es por ello, que se han aplicado las soluciones descritas en el apartado 5.2.1 para que podamos tener un corpus lo más homogéneo posible.

5.2.2.3 Fechas de las contribuciones

Debido a que no siempre se siguen las instrucciones de la guía ReDIF que proporciona RePEc, tenemos fechas en diferentes campos y a su vez en diferentes formatos. Por ejemplo, tenemos los siguientes:

- YYYY-MM-DD
- YYYY-MM
- YYYYMMDD
- Etc...

Para solucionar este problema, se han valorado varias opciones, las cuales describimos a continuación:

- Aplicar una serie de patrones de fechas, los cuales nos permiten obtener la fecha en concreto que buscamos. Esta solución es aparentemente “más limpia”, pero puede llevarnos a descartar alguna fecha que no tengamos cubierta con su correspondiente patrón.
- La segunda opción que se ha valorado corresponde en aplicar una simple expresión regular, mediante la cual buscamos 4 números de 0 a 9 consecutivos, con lo cual nos permitirá obtener el año de la fecha que estamos tratando. Esta solución se describe como “menos limpia” que la anterior por el hecho de que en formatos como los siguientes no podremos distinguir entre el conjunto de día y mes (DDMM) y el año en solitario (YYYY):
 - DDMMYYYY

- YYYYMMDD
- La última opción valorada es simplemente una extensión de la anterior, en la cual una vez aplicada la expresión regular y obtenidos los 2 valores de 4 cifras, hay que establecer un rango de años disponibles entre los que se encuentre la cifra obtenida, es decir, si tenemos el siguiente caso: 02032018 obtendremos por un lado el valor 0203 y por otro el 2018, si establecemos un rango de años, desde 1700 hasta el año actual (2019), podemos saber a ciencia cierta que el año es la segunda cifra, no obstante puede haber casos en los que nos encontremos con falsos positivos, como por ejemplo este caso: 20082018, cuya traducción correcta es el 20 del 08 de 2018, pero debido a aplicar esta solución, no seríamos capaces de distinguir entre el 2008 y el 2018.

Otro de los problemas relacionados con esto es que en algunos casos (normalmente en los *Articles*) la propiedad se llama “*Year*” y en otros como en los *Paper* se llama “*Creation-Date*”.

Para solucionar esto, existen varias soluciones. La primera de ellas pasa por crear un campo único que englobe a estas 2 propiedades, de forma que únicamente se vuelque un campo “*Creation-Date*” que nos indique el año de creación del *Paper* o *Article*. Esto nos permitirá realizar un proceso de análisis más preciso, ya que no deberemos preocuparnos de saber que campo utilizar en cada momento.

5.3 Almacenamiento y consulta

Para facilitar el almacenamiento y acceso a RePEc, se ha utilizado Elasticsearch⁹, el cual es un servidor de búsqueda basado en el motor Lucene¹⁰, dicho motor nos permite realizar consultas de forma indexada. El servidor nos provee de un API REST¹¹ al cual nos podemos conectar para realizar tanto peticiones de inserción como de búsqueda.

Elasticsearch es propiedad de la empresa Elastic, la cual tiene otros productos como Kibana¹² (dashboards) o Logstash (transformación y centralización de datos) entre otros productos. Entre todos ellos conforman lo que ellos llaman la *Elastic Stack*, en la cual

⁹ Elasticsearch: <https://www.elastic.co/es/products/elasticsearch>

¹⁰ Lucene: <https://lucene.apache.org/core/>

¹¹ REST: https://en.wikipedia.org/wiki/Representational_state_transfer

¹² Kibana: <https://www.elastic.co/es/products/kibana>

Elasticsearch es el centro de la misma, ya que es el encargado de almacenar, indexar y proveer dichos datos a las demás aplicaciones de la *Elastic Stack*.

La arquitectura que hemos montado en torno a Elasticsearch es la siguiente:

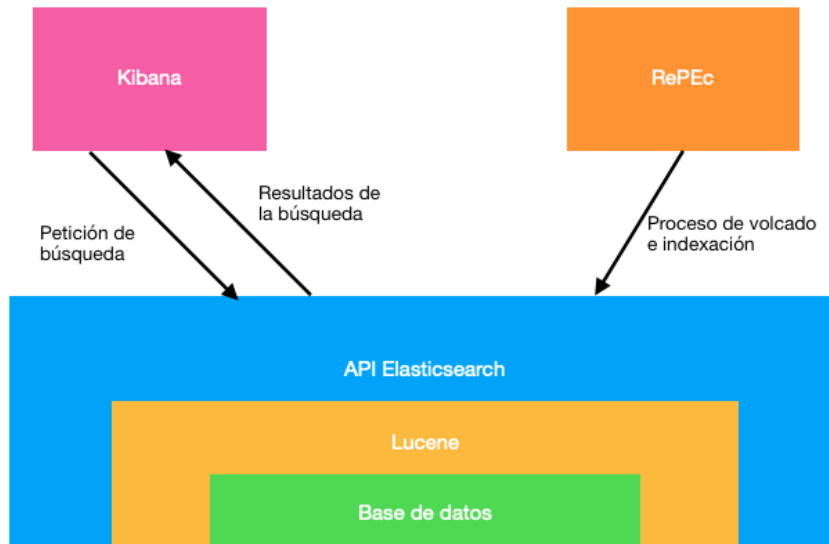


Figura 2 - Arquitectura realizada con Elasticsearch

Como podemos ver en la Figura 2, la arquitectura consta de 3 capas (Base de datos, motor de indexación y API de servicios). Cada una de ellas trabaja sobre la capa inmediatamente inferior, es decir, el motor Lucene trabaja sobre la Base de datos (indexando y realizando las búsquedas pertinentes) y el API de servicios sobre el motor de indexación (trasladando la llamada que contiene los criterios de búsqueda a dicho motor).

Tanto para almacenar documentos (podemos verlo reflejado en la figura como el “Proceso de volcado e indexación”) como para realizar las búsquedas y visualizaciones (en la figura queda reflejado en la parte de Kibana, el cual se explicará en el punto 6.1), se debe trabajar sobre el API de servicios.

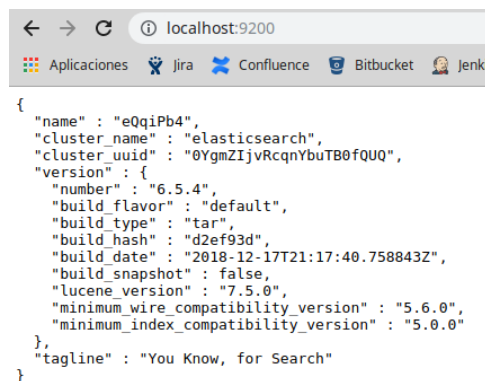
Este sistema de almacenamiento trabaja con lo que ellos llaman documentos JSON, que no son más que estructuras JSON almacenadas en la Base de Datos. La estructura de almacenamiento de este se divide en índices, los cuales nos indicarán el campo en el que nos estamos moviendo, en nuestro caso contribuciones de carácter económico (*economy*). El siguiente paso en esta estructura es el tipo de documento. Aquí indicamos los posibles tipos de documentos almacenados, en nuestro caso y por simplificar las búsquedas, el tipo siempre será *article* (es el nombre del tipo que le otorgamos a todas las contribuciones

volcadas, bien sean *Paper, Article, Software, etc...*, ya que las búsquedas las realizaremos con el campo *Template-Type*, el cual nos acotará los documentos sobre los que realizaremos la búsqueda).

Para la puesta en marcha de un servidor Elasticsearch debemos descargar los archivos de binarios desde la página web de la compañía¹³. Una vez descargados debemos extraer los archivos del ZIP en la carpeta en la cual queremos tener funcionando el servidor. Por último debemos ejecutar el siguiente comando en un Terminal presente en la carpeta que hemos extraído:

```
$ bin/elasticsearch
```

Este comando levantará el servidor Elasticsearch y lo dejará funcionando en el puerto 9200 de nuestra máquina local. Para comprobar que funciona correctamente abrimos un navegador en la siguiente dirección: <http://localhost:9200> tal y como se ve en la Figura 3. Dicha llamada nos devolverá un objeto JSON con las propiedades del servidor. Esto nos indica que se ha levantado sin problemas y que ya podemos trabajar con él con total libertad.



```
{
  "name" : "eQqiPb4",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "0YgmZIJvRcqnYbuTB0fQUQ",
  "version" : {
    "number" : "6.5.4",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "d2ef93d",
    "build_date" : "2018-12-17T21:17:40.758843Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Figura 3 - Comprobación de que Elasticsearch se ha ejecutado con éxito

¹³ La web de descarga es la siguiente: <https://www.elastic.co/downloads/elasticsearch>

5.3.1 APIs disponibles

Dentro de toda la infraestructura que podemos utilizar con Elasticsearch, este nos ofrece una serie de APIs para trabajar con la herramienta. Cada una de estas APIs nos provee de unos servicios diferentes. Las APIs de las cuales vamos a hablar son las siguientes:

- Documentos
- Búsqueda

Estas dos APIs son las que se han utilizado en el proyecto, por lo que son las que se van a explicar a continuación. En posteriores posibles ampliaciones del proyecto es posible que se utilicen otras APIs disponibles, las cuales debido a la acotación del trabajo quedan fuera del alcance actual.

5.3.1.1 Documentos

La primera de ellas es el API de documentos, la cual nos permite realizar operaciones relacionadas con los JSON almacenados en nuestro servidor. Podemos realizar una serie de operaciones simples (un CRUD¹⁴):

- *Index* (Creación de un documento)
- *Get* (Obtención)
- *Delete* (Borrado)
- *Update* (Actualización)

En la primera de ellas (*Index*) podemos realizar la inserción del documento proporcionando un identificador nosotros o en caso de no proporcionarse, nos lo asignará el propio servidor. Este identificador es importante mantenerlo, ya que para las 3 operaciones restantes, requieren de un identificador para realizar la acción sobre el documento.

La operación *Get* como hemos mencionado anteriormente, requiere de un id, el cual se debe de haber obtenido al registrar el documento. Con esto obtenemos el documento asociado a dicho id.

¹⁴ Son las siglas de *Create*, *Read*, *Update* y *Delete*. Un CRUD es un conjunto de operaciones básicas sobre un recurso. <https://es.wikipedia.org/wiki/CRUD>

Las dos últimas operaciones (*Delete* y *Update*), al igual que la operación anterior, requieren de un id que identifica el documento que queremos eliminar o actualizar.

Debido a que estas acciones, pueden ser pesadas al realizarlas de una en una, el servidor nos provee de una serie de operaciones que podemos realizar en bloque. Gracias a esto podemos realizar las mismas operaciones que antes pero con un conjunto de documentos. Las operaciones disponibles son las siguientes:

- *Multi Get*
- *Bulk*
- *Delete by Query*
- *Update by Query*
- *Reindex*

La primera operación nos permite realizar una obtención de forma múltiple, para ello podemos pasar un array de objetos JSON, donde cada objeto deberá contener el índice donde se quiere realizar la búsqueda, el tipo de documento y el identificador del mismo. Esta API nos devolverá el conjunto de documentos que correspondan con los criterios de búsqueda establecidos. Por defecto este tipo de obtenciones nos devuelven todos los campos, pero con una serie de parámetros que le indicamos en la URL (para todos los documentos que solicitamos) o en el cuerpo de cada objeto de la búsqueda (particular de cada uno de ellos), podemos indicarle que campos queremos que incluya o excluya de la respuesta.

La siguiente operación que nos proporciona nos permite realizar operaciones de forma masiva (*Bulk*), por lo que con una sola llamada nos permite hacer varias acciones (insertar, actualizar, borrar, etc..)

Las dos siguientes (*Delete by Query* y *Update by query*) como sus propios nombres ya nos lo indican, nos permiten realizar operaciones de borrado y actualización mediante una consulta, por lo que únicamente debemos pasar en la llamada los campos a modificar y el objeto que corresponda a la consulta de búsqueda.

La última de las operaciones del API de documentos es una de las más útiles, ya que nos permite reindexar un índice que ya esté almacenado en nuestra base de datos, creando de esta forma otro índice con los mismo registros. A priori puede parecer una operación sin importancia, pero en muchas ocasiones hemos tenido que cambiar la configuración de un campo, bien sea para cambiar su tipo o para cambiar el analizador o mapeador de dicho

campo, por lo que sin esta operación tendríamos que relanzar el volcado de ficheros, por lo que habría que leerlos todos de nuevo, extraer los registros y realizar de nuevo todas las inserciones. Gracias a la operación *Reindex* podemos ahorrar todo ese trabajo, de forma que la reindexación sería mucho más rápida y eficiente.

5.3.1.2 Búsqueda

El API de búsqueda es la seña de identidad del motor Elasticsearch, ya que ofrece una gran cantidad de posibilidades a la hora de realizar filtros de búsqueda. eÉta ofrece la posibilidad de realizar búsquedas en n índices diferentes a un alto rendimiento computacional, ya que cuando realizamos el volcado de los datos el propio motor se encarga de crear los índices necesarios para los procesos de búsqueda.

Gracias a la ya mencionada velocidad de búsqueda y la flexibilidad que ofrece el API a la hora de realizar los filtros, convierte a Elasticsearch en una herramienta extremadamente útil en proyectos de *BigData* y *Analytics*. Algunos ejemplos de peticiones al API de búsqueda son las siguientes:

```
GET repec/_search?q=template-type:ReDIF-Paper
```

Figura 4 - Ejemplo de petición de búsqueda simple

Esta petición realiza una búsqueda en el índice “repec”, cuyo parámetro de búsqueda es el tipo de plantilla, es decir, se quiere obtener el registro o registros que tengan como valor “ReDIF-Paper” en el campo “template-type”.

```
GET repec,repec11/_search?q=template-type:ReDIF-Paper
```

Figura 5 - Ejemplo de petición de búsqueda en múltiples índices

Este otro ejemplo, nos permite realizar la búsqueda que solicitemos en varios índices, en este caso en “repec” y “repec11”.

La potencia del API de búsqueda reside en las posibilidades que ofrece el poder enviar los criterios de búsqueda en el “body” de la petición realizada, de forma que podemos crear un objeto JSON que contenga los criterios de búsqueda. Por ejemplo, uno de los ejemplos anteriores expresado de esta forma quedaría de la siguiente manera:

```
GET repec/_search/
{
  "query": {
    "exists": {
      "field": "author-name"
    }
  }
}
```

Figura 6 - Ejemplo de petición de búsqueda con criterios en el cuerpo de la misma

En este caso en particular, se quieren obtener todos los documentos en los que existe la propiedad "author-name".

Esto nos ofrece una gran flexibilidad a la hora de crear nuestros propios filtros, ya que queda mucho más visual y sencillo de entender un objeto JSON como criterio de filtrado que una serie de parámetros en la URL, en cuyo caso sería más difícil de comprender por qué criterio/s se filtra.

5.3.1.3 Agregaciones

Las agregaciones en sí no son un API específica de Elasticsearch. Son un complemento de ayuda en los procesos de búsqueda, pero debido a la gran cantidad de agregaciones que nos permite aplicar el motor de búsqueda, se ha considerado oportuno explicarlas en un apartado en particular.

En primer lugar, las agregaciones son una serie de operaciones que aplicamos a los resultados de una búsqueda, de forma que nos permita realizar acciones o cálculos sobre los datos retornados en la consulta. Tenemos varios tipos de agregaciones, los cuales son los siguientes:

- *Metrics*
- *Bucket*
- *Pipeline*
- *Matrix*

En este trabajo, únicamente se han utilizado agregaciones de los dos primeros tipos, por lo que son los que se explicarán a continuación.

Las agregaciones de métricas (*Metrics*), nos permiten realizar cálculos sobre un conjunto de datos obtenido en una consulta previa. Gracias a este tipo de herramienta podemos calcular sumas, conteos o promedios de una forma muy sencilla y eficiente. Entre las

agregaciones de este tipo podemos encontrar cálculos estadísticos como mínimos, máximos, percentiles, desviación, etc... La forma de aplicar una agregación de este tipo sería la siguiente:

```
GET repec11/_search/
{
  "size": 0,
  "aggs": {
    "min_year": {
      "min": { "field": "creation-date" }
    }
  }
}
```

Figura 7 - Ejemplo de agregación de métricas

En este caso en particular, podemos ver como la consulta que se está realizando es sobre el índice “repec11”, al cual le aplicamos la agregación del mínimo, ya que queremos obtener el valor mínimo del campo “creation-date”, es decir, la fecha de creación más antigua. Dado que no tenemos ningún criterio de consulta, esta llamada devolvería todos los resultados y a continuación el resultado de aplicar la agregación sobre dicho conjunto de datos, pero como hemos indicado el parámetro “size” con valor 0 dicha consulta no devolverá los datos solicitados, pero sin embargo sí nos devolverá el resultado de aplicación de la agregación sobre los datos resultantes de la petición, ya que en este caso es el único dato que nos aporta valor (aquí no nos importa el conjunto de datos, solo nos importa el valor mínimo del año, ya que este nos servirá para comprobar el rango de fechas del que hemos hablado en el punto 5.2.2.3).

Otro tipo de agregaciones que tenemos disponibles son las agregaciones de “cubo” (*Bucket*), en las cuales el objetivo no es conseguir datos calculados, si no bloques de documentos que cumplan un criterio establecido. Este tipo de agregaciones son muy útiles para la obtención de grupos de palabras más empleadas, o la cantidad que aparece cierto término en un conjunto de documentos. Este tipo de agregaciones también permiten obtener resultados para la representación de histogramas o rangos.

Un ejemplo de aplicación de esta agregación es el siguiente:

```
GET repec11/_search
{
  "aggs": {
    "grupos": {
      "terms": {
        "field": "title.keyword"
      }
    }
  }
}
```

Figura 8 - Ejemplo de agregación de "cubo" o agrupación

Para este ejemplo concreto, obtenemos los resultados de la consulta y aplicamos una agregación de términos (*terms*), la cual nos proporciona una agrupación por el campo que se le indica en la petición. En este caso lo que se quiere obtener es una medida de cuántos registros se tienen agrupados por el título de la publicación. La respuesta de esta petición nos devolverá los resultados, seguidos del resultado de la agregación, los cuales nos indicarán todos y cada uno de los diferentes valores en el campo "title" y el número de ocurrencias de cada uno de ellos.

Una de las mayores bondades que ofrecen las agregaciones, es que estas pueden ser anidadas, por lo que podemos aplicar una agregación distinta al resultado de otra agregación. Esto nos permite realizar operaciones que impliquen varias variables.

La petición necesaria para aplicar una agregación anidada es la siguiente:

```
GET repec/_search
{
  "query": {
    "match_all": {}
  },
  "aggs": {
    "creation": {
      "terms": {
        "field": "creation-date",
      },
      "aggs": {
        "language": {
          "terms": {
            "field": "programming-language.keyword"
          }
        }
      }
    }
  }
}
```

Figura 9 - Ejemplo de agregación anidada sobre otra agregación

En este ejemplo que se ha ilustrado en la Figura 9, el objetivo es obtener todos los documentos, los cuales se agruparán posteriormente por fecha de creación. Por último, sobre cada uno de los resultados obtenidos aplicamos la segunda agregación y agrupamos por lenguaje de programación. Gracias a esto y a la rapidez ofrecida por el motor de búsqueda, conseguimos los resultados esperados de una manera eficiente.

5.4 Volcado de datos en Elasticsearch

Lo primero que debemos hacer es realizar la inserción del corpus de estos artículos científicos. Para ello deberemos realizar una petición POST con cada uno de los artículos a insertar. El contenido de la petición serán los datos extraídos de los archivos ReDIF. Aquí tenemos un ejemplo de cómo sería una petición de inserción:

```
{
  "Template-Type" : "ReDIF-Paper 1.0",
  "Author-Name" : "Stefan Wrzaczek",
  "Title" : "The Reproductive Value as Part of the Shadow Price of Population",
  "Abstract" : "The reproductive value (see Fisher 1930) arises as part of the shadow pri...",
  "Length" : "10 pages",
  "Creation-Date" : "2011-04",
  "File-URL" : "http://www.oeaw.ac.at/fileadmin/subsites/Institute/VID/PDF/Publica...",
  "Keywords" : "Reproductive value, distributed optimal control theory, McKendrick equati...",
  "Handle" : "RePEc:vid:wpaper:1106"
}
```

Figura 10 - Ejemplo de documento volcado

Se ha utilizado como fachada de esta petición la librería Elasticsearch en Python¹⁵, la cual nos provee de un cliente que nos realiza dicha llamada al servidor de Elasticsearch.

¹⁵ Página principal de la librería: <https://elasticsearch-py.readthedocs.io/en/master/>

5.5 Datos cuantitativos del corpus

Una vez volcado el corpus en Elasticsearch, se han realizado una serie de medidas para evaluar la riqueza del corpus, los resultados obtenidos nos indican:

- Cantidad de archivos volcados → 748156 ficheros
- Cantidad de archivos excluidos → 566 ficheros (debido a que no se han podido procesar por el *encoding*)
- Cantidad de contribuciones volcadas → 2286035 contribuciones
- Tiempo empleado en el volcado → 27528 segundos → 7,65 horas

Una vez volcado el corpus, comprobamos cuantas contribuciones tenemos de cada tipo, para ello, mostramos el siguiente gráfico :

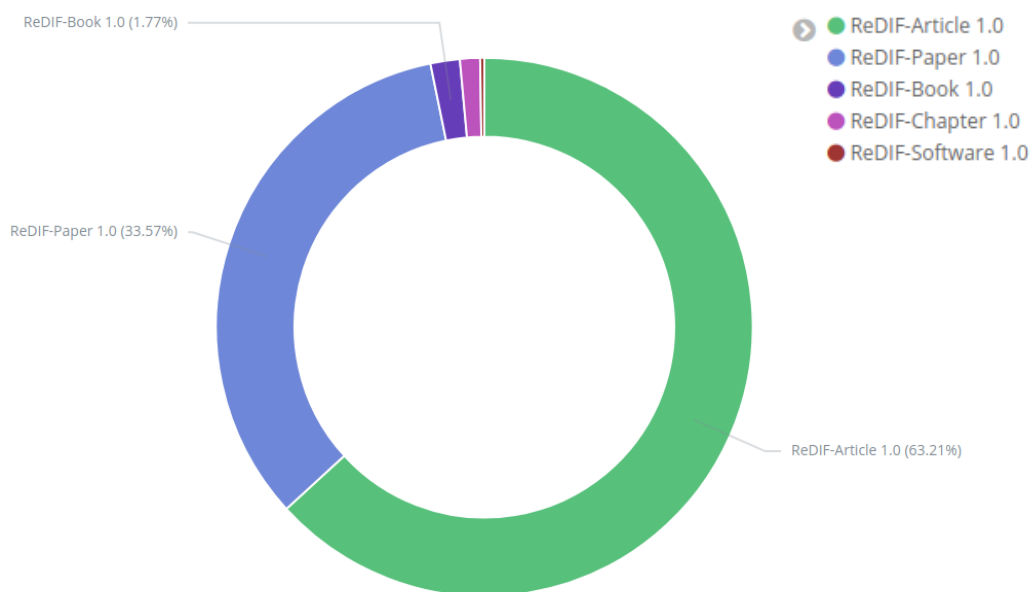


Figura 11 - Distribución de los diferentes tipos de contribuciones

Como podemos ver la gran mayoría de contribuciones son del tipo *ReDIF-Article* y *ReDIF-Paper*, entre las dos ocupan más del 95% de toda la base de datos de RePEc, pero como podemos ver hay otro tipo de contribuciones como son las del tipo *ReDIF-Software* que pese a que son muy minoritarias podemos realizar un estudio interesante sobre ellas.

A continuación se muestra una gráfica comparativa de la evolución del tiempo de ejecución respecto al número de ficheros.



Figura 12 - Evolución temporal del proceso de volcado

Como podemos observar, aunque ambas líneas son similares, la desviación que hay entre ellas se debe a que no en todos los ficheros existen el mismo número de contribuciones, por lo que hay franjas de volcado en el que los ficheros son más simples, y otras con ficheros más densos, es por ello que no tenemos un avance más lineal.

El proceso de volcado se ha realizado en una máquina con las siguientes características técnicas:

- Procesador → Intel Core I7
- Memoria RAM → 32 GB
- Disco duro → 256 GB SSD
- Sistema Operativo → Linux Mint 18

Dado, que es un proyecto con una gran cantidad de datos, se ha considerado incluir en los datos técnicos el tamaño del disco duro, así como su tipo, ya que gracias a la velocidad de lectura/escritura de los discos duros solidos (SSD) hace que el tiempo de procesamiento se reduzca en gran medida.

En un principio se intentó realizar los volcados de datos en una máquina con las siguientes características técnicas:

- Procesador → Intel Core I5
- Memoria RAM → 8 GB
- Disco duro → 500 GB SSD
- Sistema Operativo → Mac OS High Sierra

El principal problema encontrado aquí y por lo cual no se pudo realizar el volcado completo es debido a la memoria RAM, ya que en muchos casos el volcador cargaba los ficheros en memoria y dejaba el SO bloqueado, por otra parte debido que se tiene un procesador de menor capacidad de procesamiento los tiempos de ejecución hubieran sido mayores.

6. Análisis

Respecto al proceso de análisis, se han planteado dos casos de estudio, los cuales son los siguientes:

- El camino del *working paper* a publicación
- Análisis de software existente

El primero de ellos consiste en realizar un cálculo del número medio de años que tarda el *working paper* en convertirse en una publicación (*Article*), y la cantidad media de *Papers* que son publicados en repositorios antes de que sean publicados en una revista.

El segundo caso de estudio, consiste en realizar un estudio sobre qué lenguajes de programación son los más empleados en los tipos *ReDIF-Software*, qué *Keywords* son los más empleados en las contribuciones y qué sitios web son los más utilizados para albergar dicho contenido.

Para ello se han utilizado medidas estadísticas como la media aritmética y la desviación estándar. Por otro lado, haciendo uso de la herramienta Kibana de la corporación Elastic, se han realizado visualizaciones de los datos almacenados en Elasticsearch.

6.1 Kibana

Esta herramienta es propiedad de la corporación Elastic, la cual, como hemos mencionado anteriormente, es propietaria del motor Elasticsearch. Gracias a que ambas herramientas son parte de la misma arquitectura de trabajo en proyectos de *BigData* y *Analytics*, ambas se integran a la perfección, por lo que no es necesario realizar ninguna configuración especial, salvo que para futuras puestas en producción, en cuyo caso habría que hacer unas pocas configuraciones de seguridad.

Esta herramienta ofrece la posibilidad de explorar y visualizar los datos almacenados en los índices que tenemos en Elasticsearch. También permite realizar dashboards compuestos por las visualizaciones que desarrollemos a partir de nuestros datos. Para ello ofrece una interfaz muy completa:

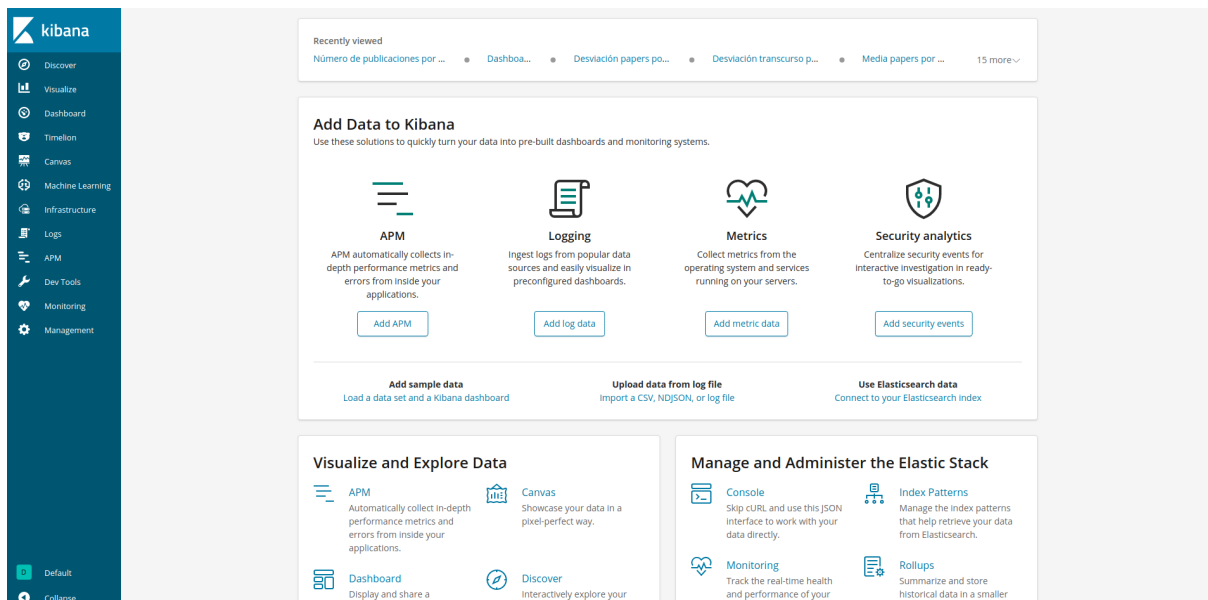


Figura 13 - Página principal de Kibana

La interfaz que vemos en la imagen contiene un menú lateral con todas las opciones que nos ofrece la herramienta. Concretamente, en este trabajo utilizaremos las siguientes opciones:

- Visualize
- Dashboard
- DevTools
- Management

6.1.1 Visualize

La primera de ellas nos permite realizar visualizaciones sobre los datos almacenados. De entre todas las posibilidades que nos ofrece esta sección, la que más se va a utilizar es el gráfico de barras verticales, ya que nos ofrece una visión cuantitativa de las variables que queramos medir.

En menor medida, se han utilizado otro tipo de visualizaciones como la métrica, la cual nos ofrece un número que indica la cantidad de registros que cumplen la búsqueda establecida, o el gráfico circular, el cual nos sirve para observar la proporción de registros que tenemos agrupando por un campo en concreto.

El resto de visualizaciones que nos ofrece la herramienta son las siguientes:

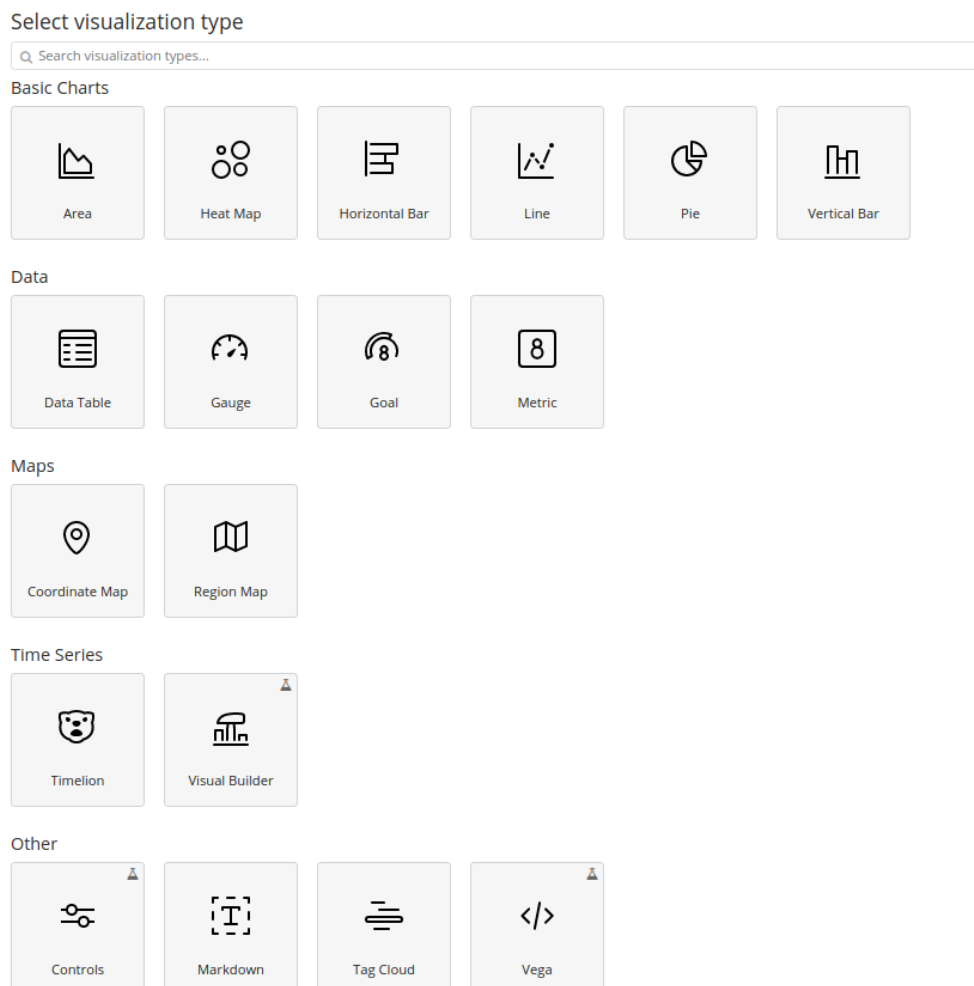


Figura 14 - Ilustración de las diferentes visualizaciones que ofrece Kibana

Como podemos ver, no ofrece una amplia variedad de visualizaciones. No obstante, aquellas que ofrece son suficientes para el desarrollo del trabajo. Se explicará el funcionamiento de las empleadas más adelante cuando se explique el desarrollo y visualización de cada uno de los casos de estudio propuestos.

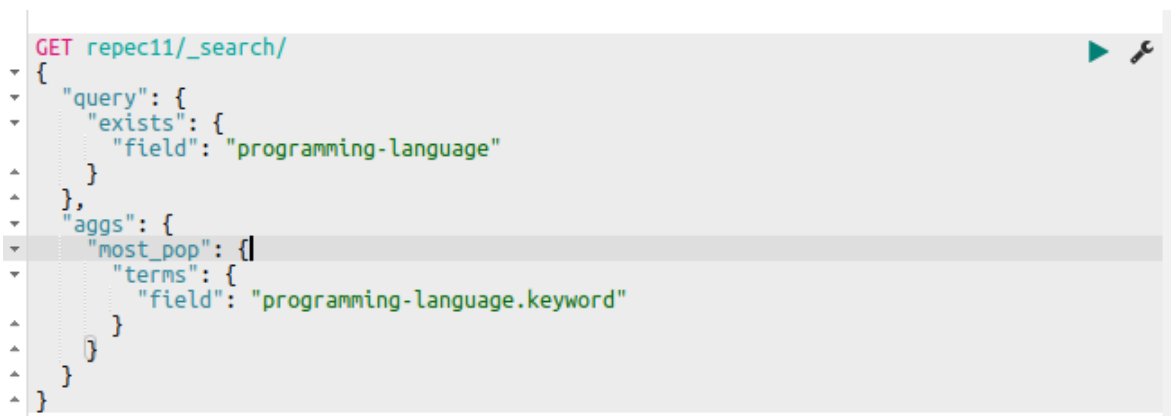
6.1.2 Dashboard

Esta sección nos permite utilizar las visualizaciones creadas previamente para poder verlas todas en una misma vista, de forma que podemos tener una visión global de los resultados obtenidos durante la fase de análisis.

Estos Dashboards que diseñemos se pueden configurar para que apliquen filtros a las visualizaciones incluidas en él, de forma que si tenemos un par de visualizaciones que nos indican datos de forma global, podemos aplicar un filtro que haga que las visualizaciones se recalculen de forma automática, con lo que podemos obtener gráficos dinámicos capaces de darnos tanto una visión global de los resultados como una más específica de un conjunto de datos en concreto.

6.1.3 Dev tools

Aquí podemos realizar peticiones de prueba al servidor Elasticsearch que queramos obtener de manera rápida. De esta forma no es necesario el uso de ningún cliente Http ni tampoco de un cliente que realicemos nosotros. La interfaz para poder realizar todas y cada una de estas peticiones es la siguiente:



```
GET repec11/_search/
{
  "query": {
    "exists": {
      "field": "programming-language"
    }
  },
  "aggs": {
    "most_pop": {
      "terms": {
        "field": "programming-language.keyword"
      }
    }
  }
}
```

Figura 15 - Consola de las DevTools que ofrece Kibana

Como podemos ver en la imagen, se pueden realizar las operaciones que necesitemos indicando el verbo Http (en este caso es "GET") a utilizar y el índice donde buscar, el cuerpo de la petición se hará conforme a lo que nos indique el API que queramos utilizar en cada momento.

6.1.4 Management

Esta interfaz es la que menos hemos utilizado, ya que nos provee de opciones para configurar la herramienta, que en este caso no vamos a utilizar, algunas de las funcionalidades que ofrece este interfaz son:

- Configuración de accesos
- Manejo de índices
- Reportes
- Etc..

Concretamente sólo hemos utilizado la segunda opción ya que para la realización de visualizaciones, es necesario definir un patrón de índice, esto nos permite englobar varios índices que compartan alguna información que nos interese y de esta forma realizar la visualización en torno a estos índices.

6.2 Proceso de análisis

Como hemos mencionado previamente, se han establecido dos casos de estudio principales, los cuales pasaremos a explicar detalladamente a continuación. En ambos casos de estudio se ha utilizado la misma fuente de datos (el índice creado previamente durante el proceso de volcado), para que de esta forma los resultados que obtengamos tengan correlación entre sí.

6.2.1 El camino del *working paper* a publicación

6.2.1.1 Objetivos del estudio

Los objetivos a abordar en este caso de estudio se basan en el estudio de los *working papers* hasta que se convierten en publicación, pero ¿Que es un *working paper*? ¿Que lo hace convertirse en publicación?.

Un *working paper* ó *paper* simplemente es una versión preliminar de un artículo, con la diferencia que este puede estar publicado en cualquier medio que no sea una revista (repositorio privado/público, etc...). Mientras un *paper* no quede publicado en una revista científica, no será considerado como artículo. Otra de las características que otorga la conversión de *paper* a *article*, es que una vez completada la publicación es una revista, esta le otorga un identificador único conocido como DOI (*Digital Object Identifier*).

Realizar este estudio nos permitirá saber en qué repositorios los *Papers* llegan a *Article* mas rápido, lo cual aportará una medida de calidad de ese repositorio. Para ello el objetivo principal de este caso de estudio es calcular el promedio del transcurso de años desde que se publica el primero de los *working papers* hasta que alcanza la publicación en una revista científica, es decir, cuando se convierte en *Article*. Para que se puedan comprobar los resultados con un mayor grado de seguridad, se ha decidido realizar los cálculos a nivel global (para todas las contribuciones que tengamos volcadas), para que así se pueda tener una visión general de las contribuciones, y a nivel de repositorio (calculando cada una de estas medias por repositorio).

Otro de los objetivos es realizar un estudio similar al anterior, pero en este caso el criterio de evaluación es el número de *papers* que se han publicado antes de convertirse en *Article*. De igual manera que el caso anterior, el objetivo es calcular el promedio a nivel general para todas las contribuciones y los promedios a nivel de repositorio.

6.2.1.2 Implementación

Para realizar el estudio, el primer paso era realizar una agrupación por el título de la contribución, de forma que obtengamos un diccionario de tipo *<String, Lista>*. Este diccionario estará formado por una clave que será el título del documento y un valor que será una lista con todos los documentos en los que el título se corresponda con la clave.

Una vez realizada la agrupación por título, para cada una de las agrupaciones se ha calculado el promedio del transcurso de años y del número de *papers* por *Article*.

Para el cálculo del transcurso de años, se ha ordenado la lista de documentos por año de creación (*creation-date*) de menor a mayor, por lo que nos quedaría una lista de documentos ordenada de forma cronológica. Una vez hecho esto, se ha calculado el transcurso de años entre cada par de documentos. Para ello se ha utilizado la siguiente expresión:

$$t = y_{i+1} - y_i$$

Donde t es el transcurso de años, y_i es el año de creación del documento actual e y_{i+1} es el año de creación del siguiente documento en orden cronológico.

Mientras vamos calculando estos valores, los vamos sumando para calcular el promedio. Una vez hemos calculado cada uno de los valores y los hemos sumado, debemos dividir el resultado obtenido por el número de *papers*, ya que el *Article* debemos dejarlo fuera (el promedio que nos interesa es el del número de *papers*, por lo que si no excluimos el *Article* lo estaríamos contando cómo un *paper* más).

Un ejemplo de cálculo de transcurso de años es el siguiente:

Tenemos un documento que ha pasado por 3 estados (2 *papers* y la publicación), con años de creación 2015, 2016 y 2018 respectivamente. Por lo tanto, debemos sacar los 2 transcurros que se han producido (de 2015 a 2016 y de 2016 a 2018), es decir 1 y 2 años. Una vez obtenidos, debemos calcular el promedio de la siguiente forma:

$$\underline{x} = \frac{1 + 2}{2} = 1.5 \text{ años de media}$$

Una vez hemos calculado los promedios del transcurso de años,¹⁶ debemos ir acumulándolos para calcular el promedio global de cada repositorio, para lo cual sumamos todos los promedios y dividimos entre el número total de grupos que hemos obtenido. De esta forma obtenemos el valor general para todos los documentos de ese repositorio.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Como hemos comentado previamente, otro de los objetivos de este caso de estudio es el cálculo del promedio del número de *papers* publicados por cada *Article*. Para ello empleamos el mismo proceso que hemos realizado para el cálculo del promedio del transcurso de años, pero en este caso aplicando de nuevo la fórmula del promedio que hemos mostrado, vamos acumulando el número de *papers* y lo dividimos entre el número

¹⁶ En economía los tiempos de publicación se miden en años, a diferencia de otras disciplinas científicas (concretamente, matemáticas, física, química, ciencias naturales y de la salud e ingeniería)

de grupos que hemos obtenido al agrupar por título, de manera que el valor que obtenemos es el promedio de *papers* que se publican por cada *Article*.

Una vez obtenidos los valores anteriormente explicados, se ha decidido calcular la desviación estándar de cada una de las métricas anteriores, de forma que podamos saber observando este valor como se distribuyen los datos.

La desviación estándar, es la medida estadística que nos indica el grado de dispersión del conjunto de valores a estudiar. El cálculo de esta medida se realiza con la siguiente fórmula:

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$$

donde s es el valor de la desviación estándar, x_i es el valor actual, \bar{x} es el valor del promedio que hemos calculado anteriormente y N es el número total de elementos en el grupo de valores. Esta medida se define como la raíz cuadrada del sumatorio del cuadrado de cada uno de los elementos del rango de valores menos el valor de la media, todo ello dividido entre el número total de elementos menos 1.

Este valor de desviación estándar se ha calculado para cada uno de los promedios anteriores:

- Transcurso de años desde el primer *paper* hasta convertirse en *Article*
- Número de *papers* por *Article*

6.2.1.3 Resultados obtenidos

Como hemos indicado previamente, se ha realizado el cálculo de dos medidas estadísticas, cuyos resultados obtenidos son los siguientes:

Promedio del transcurso de años desde el primer *paper* hasta el *Article*

En este apartado el resultado que hemos obtenido ha sido de **1,61** años, lo cual nos indica que normalmente el tiempo de publicación a *Article* es de entre 1 y 2 años, dentro de todo este tiempo entran períodos de envío, revisiones del documento y correcciones del mismo, entre otros aspectos.

Cabe destacar que para el cálculo de esta medida, se han excluido todos los documentos que se han publicado directamente como *Article* (mas del 70% del total de *Articles*), es decir, aquellos que una vez agrupados los documentos por título, únicamente se ha encontrado una ocurrencia de tipo *ReDIF-Article*. Se ha podido observar que en el caso de mantener estos documentos, el promedio del transcurso de años desciende hasta los **0,06** años, lo cual indica que hay un gran número de documentos que han sido publicados directamente como *Article*.

De la misma forma que en el caso anterior, se ha realizado el mismo cálculo pero a nivel de repositorio, de esta forma podemos ver en qué repositorios se suelen publicar más tarde y en cuales mas temprano.

Una vez hechos los cálculos hemos generado tres visualizaciones en Kibana: una con los repositorios donde se tarda menos, otra con los repositorios donde se tarda más en llegar a publicación y una última que representa la proporción de documentos por franja de años. La primera que vamos a mostrar en la de los diez repositorios en los que se tarda menos en publicar:

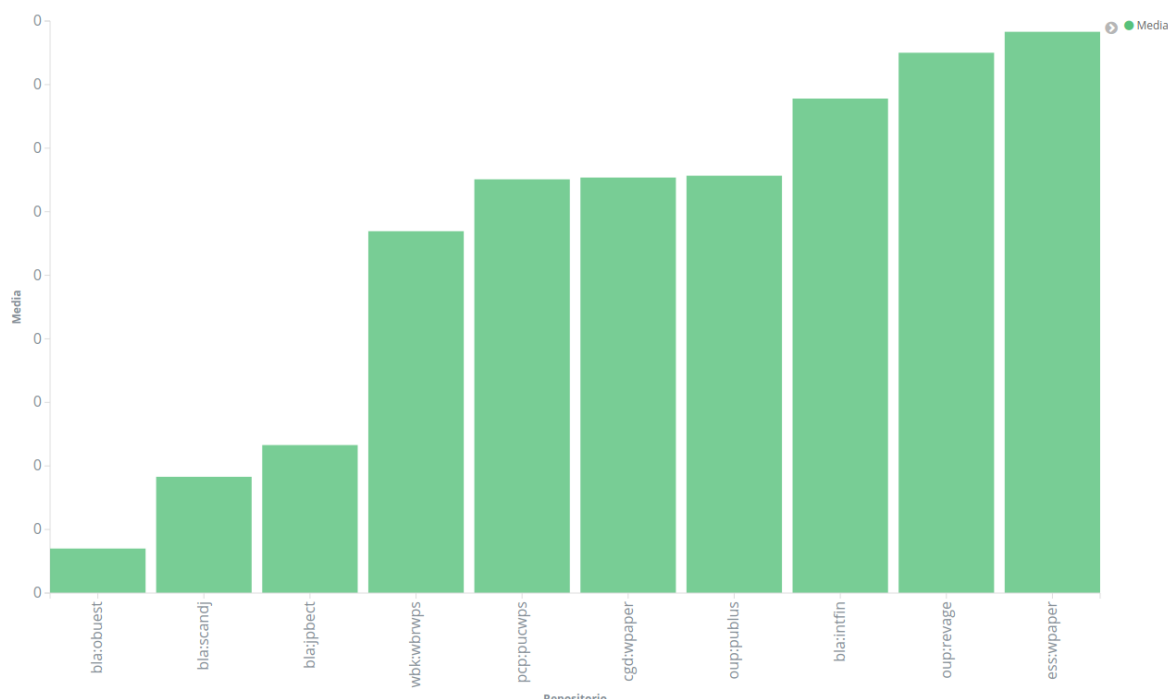


Figura 16 - Ilustración de los 10 repositorios en los que menos se tarda en llegar a Article

Como podemos ver, en prácticamente todos los repositorios del gráfico el tiempo de publicación es muy cercano a cero (el valor más a la izquierda es de 0.000014 años y el

de más a la derecha es de 0.00018 años). Hay que destacar los tres más bajos, ya que su tiempo es tan cercano a cero que el gráfico lo marca directamente como tal. Estos repositorios son los siguientes:

- bla:obuest (0.000014 años) → *Oxford Bulletin of Economics and Statistics*
- bla:scandj (0.000037 años) → *Scandinavian Journal of Economics*
- bla:jpbect (0.000047 años) → *Journal of Public Economic Theory*

El nombre de cada uno de estos repositorios lo hemos sacado del propio FTP del RePEc, ya que a nivel de repositorio en la estructura de directorios de RePEc, existe un archivo con el nombre del repositorio que contiene la información de los mismos. Por ejemplo: *blaseri.rdf*

A continuación se muestra un gráfico ilustrando los 10 repositorios en los que se suele tardar más en publicar:

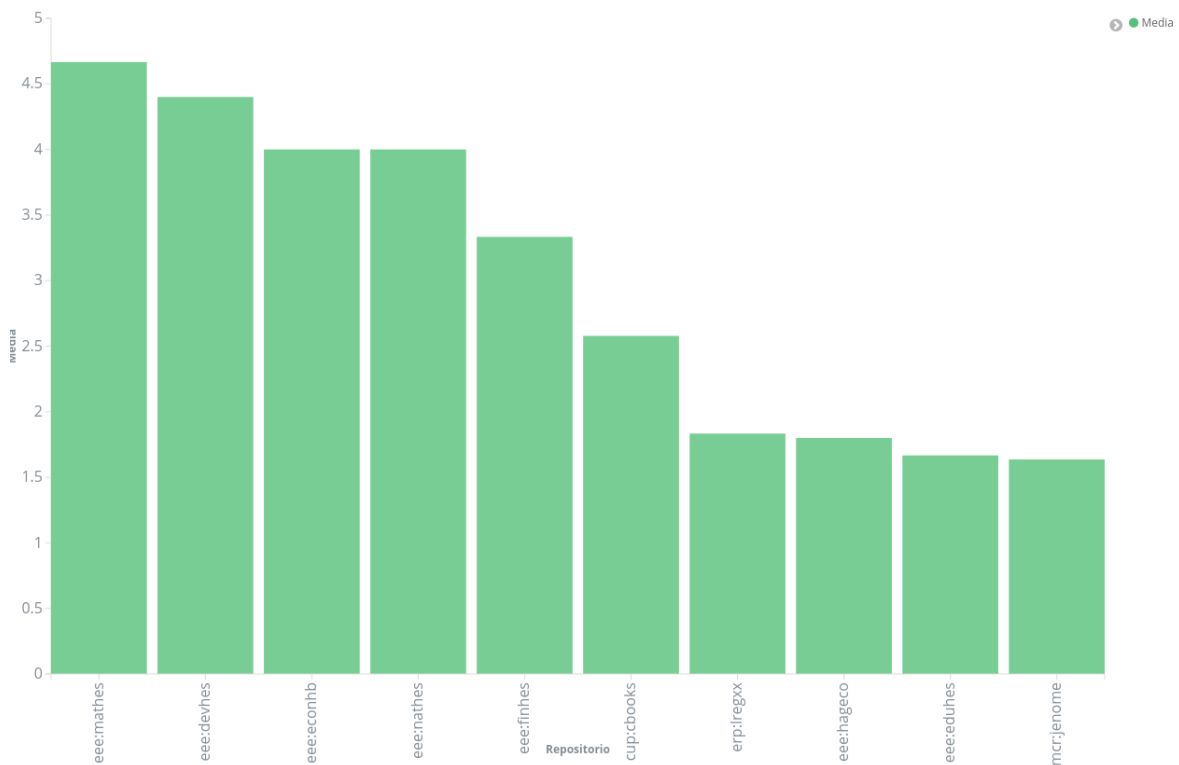


Figura 17 - Ilustración de los 10 repositorios en los que se tarda mas tiempo en llegar a Article

Como podemos ver en la imagen, el valor máximo de publicación es de poco más de **4,5** años, como hemos visto anteriormente, la media de publicación se establecía en torno a **1,6** años, lo cual nos hace ver que los valores están bastante repartidos entre los

documentos, pero esto lo podremos comprobar gracias a la desviación estándar que hemos calculado y que pasamos a mostrar.

Los tres repositorios en los que se suele tardar más en publicar, son los siguientes:

- eee:mathes (4.67 años) → *Handbook of Mathematical Economics*
- eee:devhes (4.4 años) → *Handbook of Development Economics*
- eee:econhb (4 años) → *Handbook of Econometrics*

La desviación estándar que hemos calculado en base a la media del transcurso de años, nos indicado con un valor de **3'2124** que el conjunto de valores es bastante disperso, por lo que la gran mayoría de documentos no han tenido un transcurso muy cercano a la media aritmética. Esto lo podemos deducir ya que la desviación estándar nos indica el grado de dispersión del conjunto de documentos, ya que cuanto más cercano sea su valor a cero más concentrados están los valores en torno a la media. En este caso en particular el valor es bastante elevado para el valor de la media que hemos obtenido, por lo que podemos deducir que el conjunto de datos tiene valores en una gran cantidad de rangos. Esto lo podemos ver descrito en el gráfico anterior, en cuyo caso podemos ver que tenemos tiempos de publicación bastante elevados en algunos casos.

Como podemos ver en los cálculos por repositorio, hay algunos repositorios en los que la media está por encima de los 2 años. Si atendemos al valor de la desviación estándar podemos ver que dado que su valor no es muy cercano a cero la gran mayoría de estas contribuciones tienen un tiempo de publicación bastante variable.

A continuación vamos a mostrar un gráfico que nos indica la proporción de documentos por franja de tiempo.

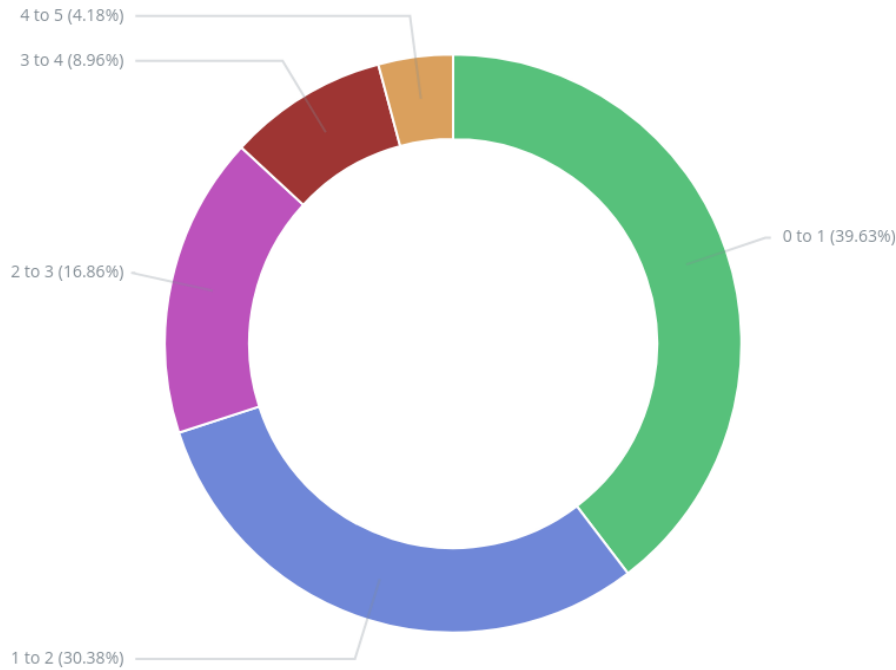


Figura 18 - Distribución de los artículos por franjas de tiempo de publicación

Como podemos ver en la Figura 18, casi el 40% de los documentos están publicados en el mismo año en el que se publica el primer *Paper*, pero debido a que el 60% restante está en un tiempo superior al año hace que el valor de la media (recordemos que es de 1.6 años) aumente. En el 60%, que representa a los documentos publicados como *Article* en años posteriores a la primera publicación, vemos que el 50% representa a los que se publican en el año siguiente, es por ello que el valor de la media a quedado encuadrado en este rango.

Promedio de la cantidad de *papers* hasta la publicación

En es este segundo punto del caso de estudio actual, se quería obtener el número medio de *papers* que son publicados antes de conseguir una publicación como *Article* en alguna revista científica. Al igual que hemos realizado anteriormente, este cálculo se ha realizado tanto a nivel general para todos los documentos como a nivel de repositorio.

A nivel general, el número medio de *papers* es de **1'3851**, por lo que sabemos que casi con toda seguridad por cada *Article* publicado tenemos como mínimo un *paper* asociado. En este caso, como podremos comprobar más adelante, el grupo de documentos es mucho

más disperso que en el caso del transcurso de años desde el primer *paper* hasta la publicación.

De la misma forma que en el caso anterior, se han descartado todos aquellos grupos de documentos que únicamente tenían un *ReDIF-Paper* una vez hecha la agrupación por título. Al igual que antes, si realizamos los cálculos con estos documentos el valor del promedio era prácticamente 0, ya que como hemos comentado previamente la gran mayoría (cerca del 70%) de contribuciones están almacenadas directamente como *Article*. En este caso de estudio, también se ha realizado el cálculo a nivel de repositorio, y de la misma manera que en el anterior, pasamos a mostrar los repositorios en los que es necesario un número menor de *papers* antes de la publicación.

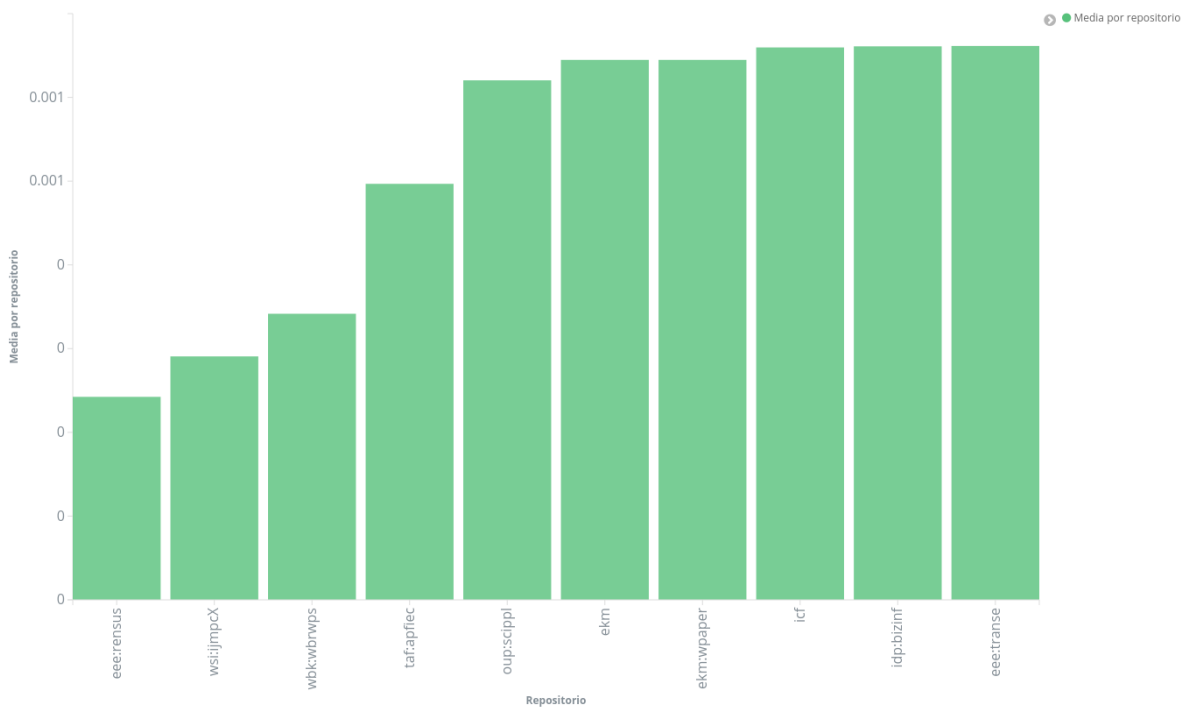


Figura 19 - Ilustración de los 10 repositorios en los que se necesita un menor número de *Papers* para conseguir publicarse como *Article*

Como podemos ver, en prácticamente todos los repositorios del gráfico el número medio de *papers* es muy cercano a cero (el valor más a la izquierda es de 0.00024 años y el valor de más a la derecha es de 0.00069 años). Hay que destacar los tres más bajos, ya que su cantidad es tan cercana a cero que el gráfico la marca directamente como tal. Estos repositorios son los siguientes:

- eee:rensus (0.00024 papers) → *Renewable and Sustainable Energy Reviews*
- wsi:ijmpcx (0.00029 papers) → *International Journal of Modern Physics C (IJMPC)*
- wbk:wbrwps (0.00034 papers) → *Policy Research Working Paper Series*

A continuación vamos a mostrar los 10 repositorios en los que es necesario un mayor número de *papers* antes de llegar a *Article*.

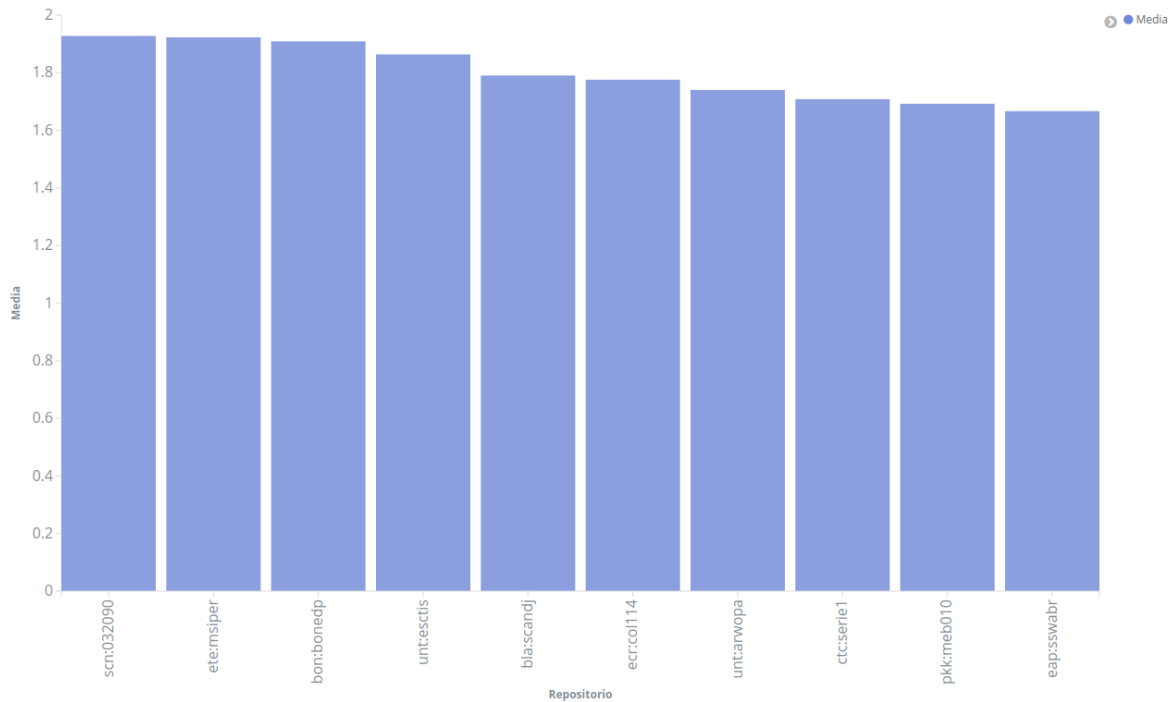


Figura 20 - Ilustración de los 10 repositorios en los que es necesario un mayor número Papers para conseguir una publicación como Article

Como podemos ver en el gráfico anterior, la media general por repositorio no excede los 2 *papers* antes de su publicación. Esto nos indica que las publicaciones tienen entre 1 y 2 *papers* además de su *Article*. Además, como explicaremos a continuación, el valor de la desviación estándar nos confirmará que se trata de un conjunto de datos bastante concentrado en torno a la media.

Los 3 repositorios en los que es necesario un mayor número de *papers* para conseguir la publicación son los siguientes:

- scn:032090 (1.93 papers) → *Российский экономический барометр*
- ete:msiper (1.92 papers) → *Working Papers Department of Managerial Economics, Strategy and Innovation (MSI)*
- bon:bonedp (1.90 papers) → *Bonn Econ Discussion Papers*

A continuación mostramos un gráfico que ilustra la distribución de los documentos por franjas de cantidad de *Papers* por *Article*.

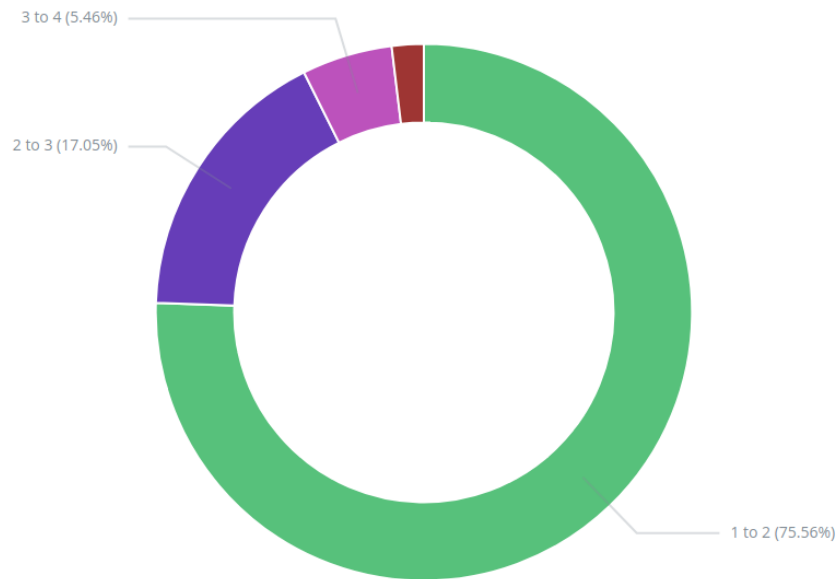


Figura 21 - Gráfico que ilustra la distribución de los documentos en franjas por cantidad de *Papers* por *Article*

Como podemos ver en la figura, el 75% de los documentos están en la franja entre 1 y 2 por cada *Article*, este dato concuerda con la media (recordemos que es de 1.3851 *Papers*) y nos confirma que la gran mayoría de documentos necesitan como mucho 1 o 2 publicaciones previas para conseguir la publicación final como *Article*. Este dato lo corroboramos a continuación con el resultado obtenido en la medida de la desviación estándar.

Como ya hemos mencionado previamente, explicaremos el valor obtenido en el cálculo de la desviación estándar, con la cual determinaremos la dispersión de los datos. En este caso en particular, la desviación obtenida ha sido de **0'8306**, como podemos comprobar en este caso de estudio el valor si es más cercano a 0 que en el caso del transcurso de años, por lo que podemos deducir que una gran cantidad de documentos tienen una cantidad de *papers* dentro de la media con un margen de error bastante pequeño.

6.2.2 Análisis de software existente

6.2.2.1 Objetivos del estudio

Los objetivos principales de este segundo caso de estudio son los siguientes:

- Lenguajes de programación más utilizados y su evolución temporal.
- *Keywords* más comunes y su evolución temporal.
- Sitios web más comunes para almacenar el contenido.

En el primero de los casos, nos dimos cuenta que existía un tipo de plantilla que es *ReDIF-Software*. Esta plantilla nos indica una contribución vinculada a un programa de software desarrollado, debido a que este proyecto se realiza para un Grado en Ingeniería Informática, nos parecía interesante incluirlo en el proyecto. El principal objetivo aquí es ver qué lenguajes de programación son los más utilizados para el desarrollo de dicho programas y su evolución a lo largo de los años.

En el segundo caso, se ha querido comprobar qué *Keywords*¹⁷ son los más empleados por los autores a nivel general, lo cual nos permitirá saber cuales son los temas más comunes a tratar entre ellos.

Por último, se ha querido comprobar qué sitios web son los más demandados a la hora de alojar un software, por lo que se mostrará la URL de los sitios web más empleados. De manera que se pueda aprovechar el proceso, también se ha decidido realizar la comprobación a nivel general de todos los documentos.

6.2.2.2 Implementación

Para la implementación de este caso de estudio se han tenido que realizar unas pequeñas modificaciones en el proceso de volcado, ya que era necesario descomponer el campo *Keywords* en los distintos tópicos que son utilizados en cada documento.

¹⁷ Definimos como *Keywords* a un conjunto de palabras o tópicos que nos indican los temas principales de la contribución en este caso.

Dicho campo nos proporciona las palabras clave como una cadena separada por un carácter especial (“,” ó “;”), por lo que debemos realizar una partición por cada uno de estos caracteres especiales y de esta forma obtener el conjunto de tópicos de cada uno de los documentos.

Se ha tenido que realizar otra pequeña modificación para volcar las URLs de los sitios web más utilizados. Para ello se ha utilizado la siguiente expresión regular:

`.*\:\/\/ (. *?) \\/`

Con esta expresión regular, obtenemos la parte de la URL que corresponde al dominio web (por ejemplo, si tenemos la siguiente URL: <https://www.ua.es/eps/grados.ppt> la expresión regular nos daría una concordancia para la siguiente subcadena: www.ua.es, que en cualquier caso es la parte que nos interesa).

Una vez realizadas dichas modificaciones, se ha procedido a realizar el volcado de nuevo. Debido a que no realizamos ningún filtrado nuevo y que solo añadimos dos campos nuevos (*keywords-splitted* y *site*) a cada documento, el proceso de volcado ha guardado el mismo número de contribuciones.

Los demás procesos de este caso de estudio se han realizado a través de la herramienta Kibana y el API de búsqueda de Elasticsearch (explicadas previamente), las cuales nos ofrecen las funcionalidades necesarias para la obtención y visualización de los datos almacenados. En este caso la funcionalidad que mayor rendimiento nos ha producido son las agregaciones (muchos de nuestros estudios se basan en agrupar y contar), las cuales nos permiten obtener conjuntos de datos con un valor en común (por ejemplo la fecha de creación → *creation-date*).

Para obtener los lenguajes de programación más empleados, se ha realizado una petición al servidor Elasticsearch solicitando todos aquellos documentos que contengan el campo “programming-language”, el cual nos indica el nombre del lenguaje de programación en el que se ha realizado el software asociado a la publicación.

Una vez realizada la petición, realizamos una agregación de términos, con la cual agrupamos los valores obtenidos por el campo “programming-language”, lo cual nos

indicará el número de ocurrencias de cada lenguaje. Concretamente se ha realizado la siguiente petición:

```
GET repec11/_search/
{
  "query": {
    "exists": {
      "field": "programming-language"
    }
  },
  "aggs": {
    "most_pop": {
      "terms": {
        "field": "programming-language.keyword"
      }
    }
  }
}
```

Esta petición nos devuelve el número de ocurrencias de cada dato, por lo que es la que utilizaremos para construir la visualización, la cual realizaremos haciendo uso de la herramienta Kibana.

Metrics

Y-Axis

Aggregation [Count help](#)

Count

Custom Label

Nº de publicaciones

[Advanced](#)

Add metrics

Buckets

X-Axis [Terms help](#)

Aggregation

Terms

Field

programming-language.keyword

Order By

metric: Nº de publicaciones

Order

Descend

Size

15

Group other values in separate bucket [?](#)

Show missing values [?](#)

Custom Label

Lenguajes

[Advanced](#)

Figura 22 - Interfaz para crear una visualización mediante agregaciones

Como podemos ver en las imágenes anteriores, se ha realizado la misma agregación que hemos mencionado anteriormente pero a través de la interfaz gráfica de Kibana, la cual una vez realizada la petición nos ilustra el gráfico solicitado que mostraremos más adelante.

Para realizar la visualización pero dividida por años, tenemos que anidar la agregación anterior a una agregación realizada previamente por el campo "creation-date". El resultado de esto nos ofrecerá un gráfico separado por años y en cada uno de ellos los lenguajes más utilizados.

De igual forma que para el lenguaje de programación, podemos realizar un estudio sobre los *Keywords* y los sitios web más utilizados siguiendo el mismo proceso, ya que el objetivo de este estudio es el mismo que para el lenguaje de programación, y en su defecto el gráfico a mostrar se creará de la misma forma.

También se han realizado dos visualizaciones adicionales para estos dos últimos casos, como nubes de palabras para poder tener de una forma más visual que el gráfico de barras.

6.2.2.3 Resultados obtenidos

Aquí mostraremos los resultados obtenidos en los principales estudios realizados en este segundo caso de “Análisis del software existente”. Para ello se ha utilizado la herramienta Kibana para realizar las visualizaciones que vamos a mostrar y explicar a continuación.

Uso de los diferentes lenguajes de programación y su evolución temporal

Los resultados obtenidos para poder comprobar el uso de los diferentes lenguajes de programación son los siguientes:

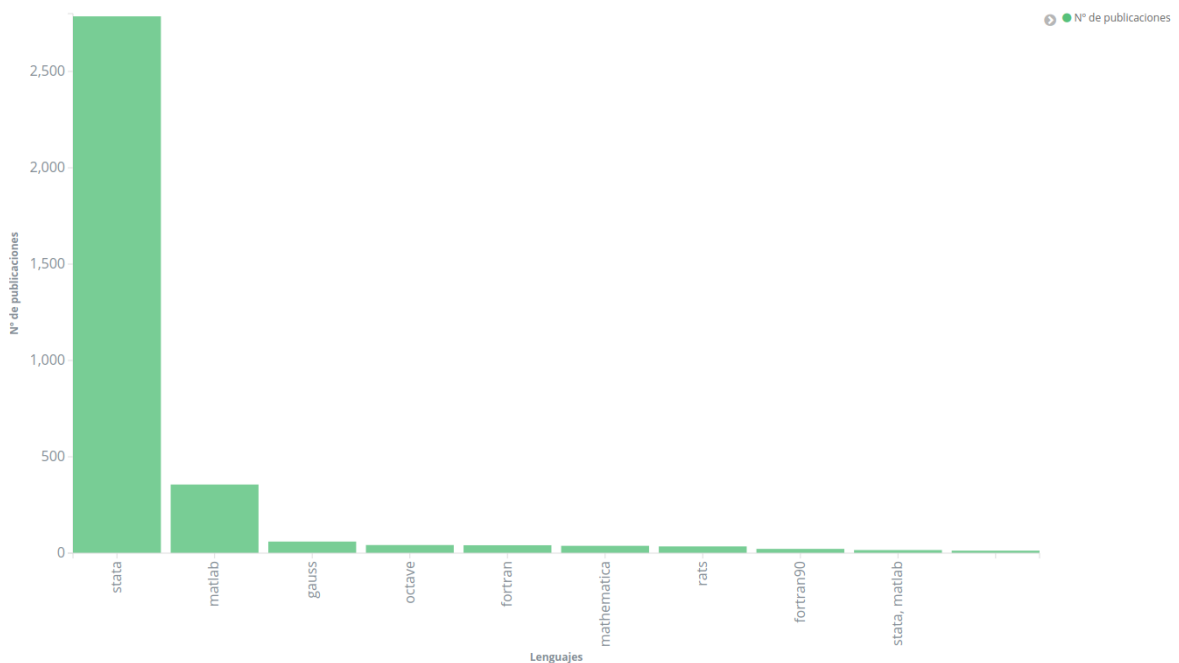


Figura 23 - Lenguajes de programación más empleados para el desarrollo de contribuciones del tipo ReDIF-Software

Como podemos observar, hay dos grandes lenguajes de programación con mayor uso, los cuales son *Stata* y *Matlab*, el primero de ellos tiene una clara ventaja respecto al resto (esto se debe a que hay una revista que se dedica únicamente a publicar software realizado con *Stata*, por lo que dispara el resultado de forma abismal).

Para poder tener una visión más clara de este estudio, se ha realizado la misma visualización pero excluyendo los resultados tanto de *Stata* como de *Matlab*, la cual nos ha quedado de la siguiente forma:

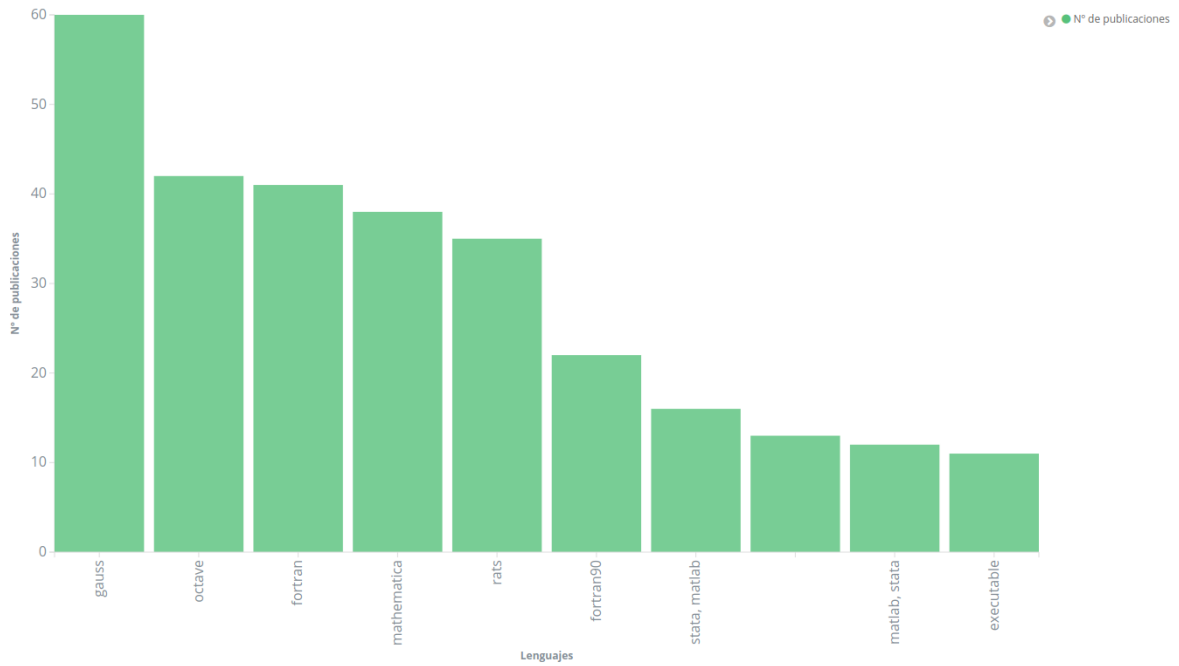


Figura 24 - Lenguajes de programación más empleados para el desarrollo de contribuciones de tipo ReDIF-Software excluyendo Stata y Matlab

Como podemos ver ahora el resultado es más equitativo, por lo que podemos ver que si obviamos los dos grandes lenguajes, tenemos cinco lenguajes que también tienen mucha presencia en los datos, los cuales son: **gauss, octave, fortran, mathematica y rats.**

Estos 5 lenguajes son los que más presencia tienen después de los dos grandes que en este caso hemos excluido.

Si realizamos esta misma comprobación (excluyendo Stata y Matlab) pero separada por años de creación, obtenemos el siguiente gráfico:

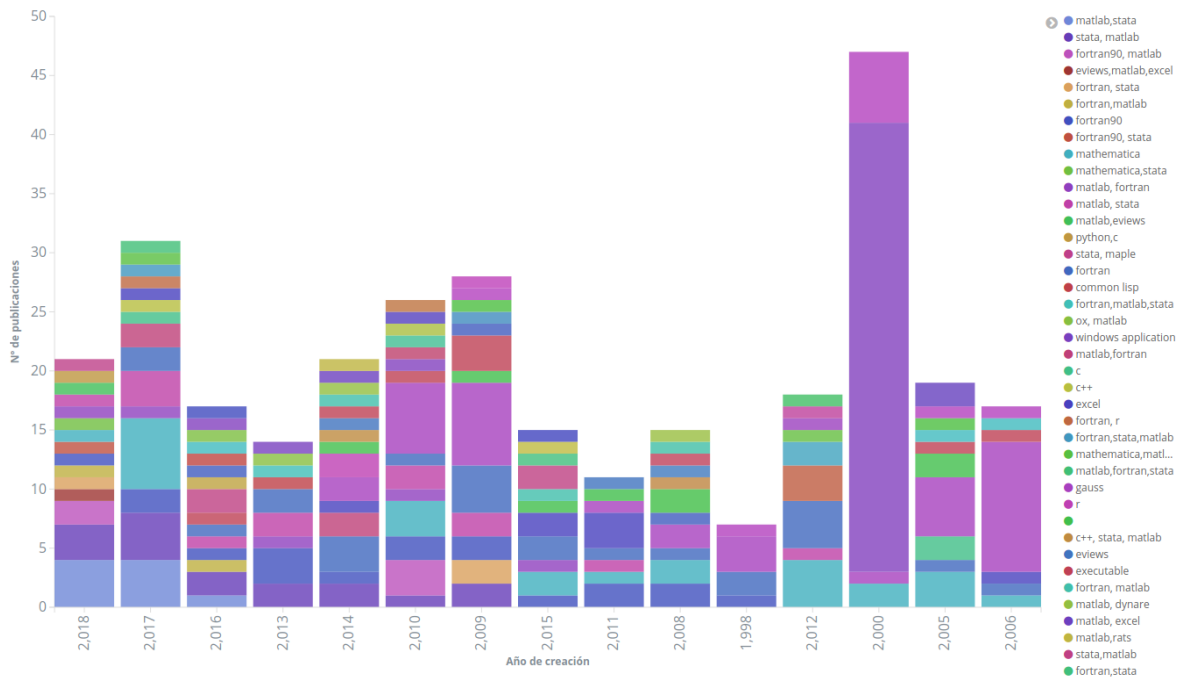


Figura 25 - Lenguajes de programación mas empleados en el desarrollo de contribuciones de tipo ReDIF-Software especificado por la fecha de publicación

Como podemos comprobar en el gráfico vemos que no hay un claro ganador en cada uno de los años, todo está relativamente equitativo, excepto en el año 2000 y 2006 en cuyo caso podemos ver que tenemos unos resultado bastante claros, los cuales son **octave** (fué utilizado en 38 de las 47 publicaciones) y **gauss** (en 11 de 17) respectivamente.

Aunque en el gráfico anterior no se muestran, cabe destacar que debido al gran número de publicaciones de *Stata* y *Matlab*, en ningún año se ha conseguido superar a estos dos lenguajes.

Keywords más utilizados y su evolución en el tiempo

En este caso, vamos a comprobar y explicar los resultados obtenidos en el estudio de los *Keywords* más populares, para ello ilustramos un gráfico de barras con la comparación de los tópicos más comunes:

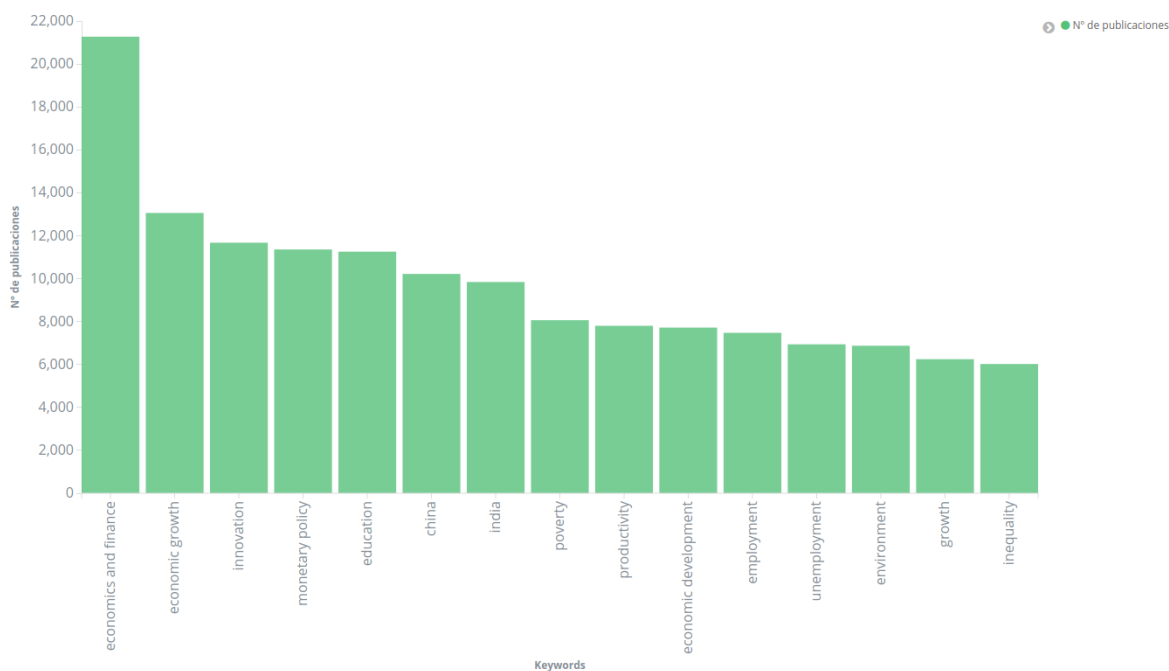


Figura 26 - Keywords mas empleadas en las contribuciones

Como podemos ver en el gráfico anterior, tenemos un tópico que destaca sobre los demás, el cual no ofrece una gran sorpresa debido al ámbito que estamos tratando en este proyecto, este tópico es: **economics and finance**. Este *Keyword* se repite en más de 20.000 documentos, después de este se encuentra un grupo de palabras bastante equitativo, las cuales están por encima de las 10.000 ocurrencias, entre ellos el más destacado es **economic growth**, por lo que podemos deducir que una gran parte de documentos hablan sobre el crecimiento económico a nivel mundial o a nivel de ciertos países.

Sobre esto podemos ver que además de los anteriores tenemos dos tópicos con nombres de países como son: **China e India**, por lo que entendemos que en muchos casos se hablará del crecimiento económico en estos países.

Otro de los aspectos a valorar, es que también se tiene el tópico **Innovation**, con lo cual podemos ver que en muchas ocasiones este crecimiento económico puede deberse a la apuesta por las nuevas tecnologías e I + D en muchos casos.

A continuación vamos a mostrar el mismo gráfico pero separado por años, por lo que podremos ver que temas son los que más se han abordado en cada año:

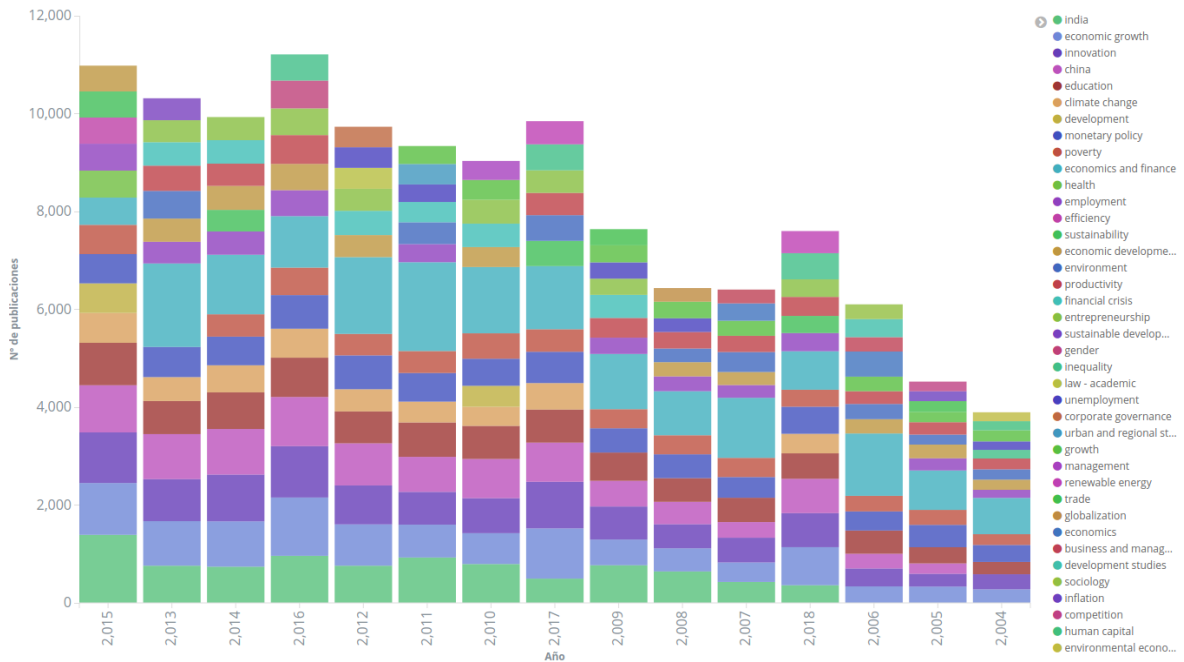


Figura 27 - Keywords mas empleados en las contribuciones especificado por la fecha de publicación

Como podemos ver, hay bastante igualdad en la división por años, pero como podemos ver hay un término que se utiliza en mayor proporción en todos los años, el cual es el mismo que hemos visto a nivel general, es decir **economics and finance**. No obstante podemos deducir que en todos los años se utilizan los mismos tópicos de forma bastante equitativa. Como hemos indicado previamente, vamos a mostrar una nube de palabras que nos indique de forma más visual el uso de los tópicos.



Figura 28 - Nube de términos que ilustra los Keywords mas empleados

Sitios web más utilizados para albergar las publicaciones

En este caso se han realizado 2 estudios, uno para las contribuciones de tipo *Software* y otro para el conjunto total de contribuciones. El primero que vamos a mostrar es el de los sitios web utilizados para almacenar el *Software*.

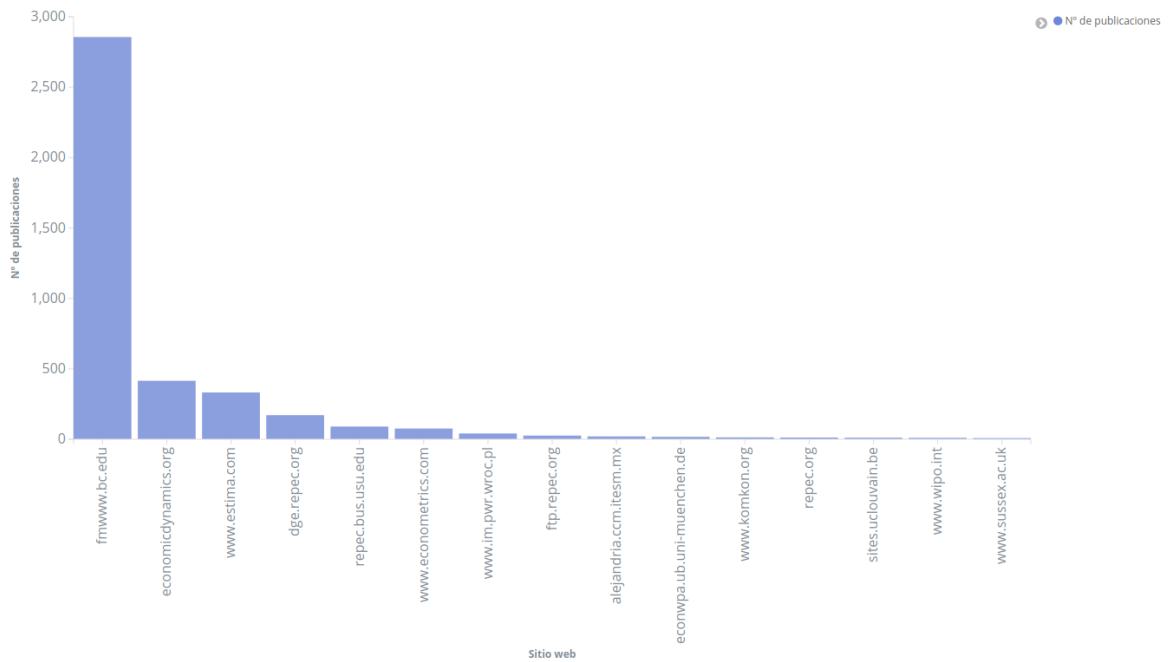


Figura 29 - Sitios web mas empleados para almacenar el software

Claramente vemos que el sitio web más empleado para almacenar el *software* es <http://fmwww.bc.edu/> con mas de 2.800 publicaciones almacenadas. Este sitio web es una plataforma del Boston College¹⁸ encargada de almacenar y dar soporte tanto a los autores como a los que consulten la contribución de *software*. Para ilustrar con mayor facilidad el dominio de este sitio web mostramos la siguiente nube de términos con los 15 sitios web más utilizados para albergar software:

¹⁸ <https://www.bc.edu/content/bc-web/bcnews.html/>



Figura 30 - Nube de términos que ilustra los sitios web mas empleados para almacenar el software

A continuación mostraremos los resultados obtenidos en el estudio realizado para la comprobación de que sitios web son los más empleados para el almacenamiento de los documentos asociados a cada contribución, pero en este caso para todos los documentos almacenados.

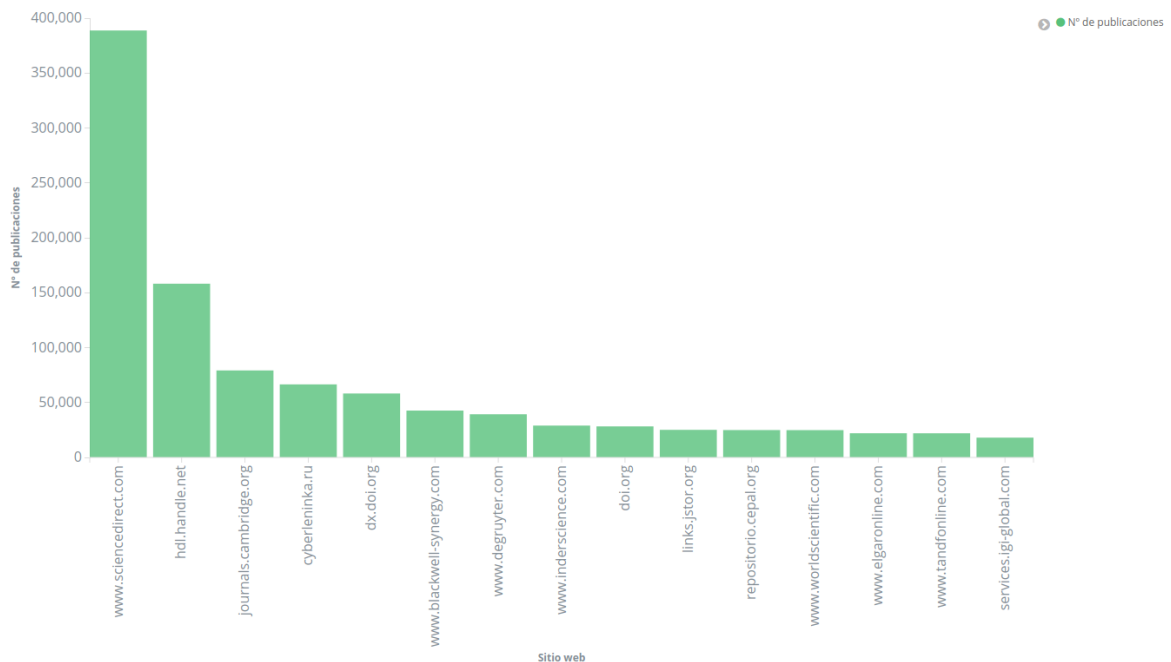


Figura 31 - Sitios web mas empleados para almacenar las contribuciones a nivel general para todos los documentos

En la gráfica anterior queda claro que hay un sitio web que predomina sobre todos los demás, el cual es el siguiente: www.sciencedirect.com. Este sitio web nos permite realizar búsquedas de documentos por autor o por palabras clave o *Keywords*. Como podemos ver este es utilizado en la gran mayoría de documentos, concretamente en más de **350.000** contribuciones. Lejos quedan el segundo y tercer sitio web, los cuales son: hdl.handle.net y journals.cambridge.org. El primero de ellos nos permite realizar búsquedas haciendo uso del campo **Handle** (recordemos que este campo ha sido utilizado previamente para obtener el repositorio en el cual se encuentra albergada la publicación y posteriormente agrupar por él para poder caracterizar los resultados del caso de estudio anterior. Véase *punto 6.2.1*). El segundo de ellos es un departamento de la universidad de Cambridge encargado de almacenar los documentos y posteriormente permitir a los usuarios que tengan acceso a ellos mediante la búsqueda de los mismos.

Al igual que en el caso anterior, a continuación mostramos una nube de términos para poder ver de forma más visual que sitios son los más utilizados por los autores para albergar el contenido de sus publicaciones.



Figura 32 - Nube de términos que ilustra los sitios web mas empleados para almacenar las contribuciones a nivel general para todos los documentos

7. Conclusiones

A lo largo de este trabajo, se ha realizado un estudio sobre un corpus de artículos científicos de ámbito económico. El objetivo principal de este trabajo ha sido aplicar técnicas de minería de datos y texto para realizar un análisis cuantitativo y cualitativo de un corpus de gran tamaño de documentos científicos de ámbito económico.

Con la realización de este análisis, se ha querido responder a dos cuestiones planteadas por expertos en el dominio y demostrar con los resultados del mismo el potencial que tiene aplicar estas técnicas sobre un corpus de gran tamaño.

Gracias a los análisis realizados, se han podido descubrir datos relevantes que van desde la media de años que tarda un *paper* en convertirse en *Article*, o qué cantidad de documentos suele ser necesaria para conseguir dicha publicación. Incluso se ha podido saber qué sitios web son los más empleados para albergar el contenido y que lenguajes de programación más empleados en las contribuciones de tipo *Software*. Este tipo de análisis no podrían haberse realizado de no ser por la existencia de este corpus de gran tamaño.

Pese a que no es un trabajo relacionado con el itinerario estudiado en el grado (Ingeniería del Software) me ha agradado realizar este estudio, ya que la temática del *BigData* y *Analytics* siempre me ha parecido muy interesante.

Todo ello me ha ayudado a adquirir nuevas competencias en el ámbito de la minería de datos y texto, la visualización de datos y el manejo de grandes volúmenes de datos, ya que estas habilidades no se trabajan durante el grado.

Por otra parte las herramientas utilizadas durante el proceso de desarrollo de este proyecto han sido de gran utilidad, ya que son herramientas de uso común en proyectos de este tipo, y su facilidad de uso hace que nos ayuden tanto en el proceso de volcado como en el proceso de análisis.

Respecto a las impresiones del trabajo realizado, cabe destacar que el proceso de análisis ha quedado bastante completo en lo que se refiere a estudio del corpus y la justificación de los resultados obtenidos. No obstante, es posible que la parte de agrupamiento por título sea mejorable mediante el uso de otros algoritmos de comparación de cadenas que nos proporcionen mayor efectividad y eficiencia (el mayor problema del método empleado, es que puede que nos produzca en algunos casos que los títulos que no están bien escritos

no se correspondan entre ellos) empleado en lo que se refiere a dicha fase del proceso de análisis.

Por otro lado el proceso de volcado ha sido bastante duro, ya que como bien hemos mencionado en el apartado 5.2.2, no todos los contribuyentes al repositorio siguen el estándar establecido por los creadores del formato ReDIF, lo cual hace que este proceso necesite una cantidad de operaciones de comparación mayor y por lo tanto un mayor tiempo de procesamiento del corpus.

Referencias

Angrist, J., Azoulay, P., Ellison, G., Hill, R. y Lu, S. F. (2017). "Economic Research Evolves: Fields and Styles." *American Economic Review*, 107 (5): 293-97.

Blank, R. M. (1991). "The Effects of Double-Blind versus Single-Blind Reviewing: Experimental Evidence from The American Economic Review". *The American Economic Review*. Vol. 81, No. 5, pp. 1041-1067

Bramoullé, Y. y Ductor, L. (2018). "Title length". *Journal of Economic Behavior & Organization*, Vol. 150, pp. 311-324.

Card, D. y DellaVigna, S. (2013). "Nine Facts about Top Journals in Economics", *Journal of Economic Literature*, Vol. 51, No. 1, 144-161.

Conley, J.P., Crucini, M.J., Driskill, R.A. y Onder, A.S. (2011). "Incentives and the Effects of Publication Lags on Life Cycle Research Productivity in Economics", NBER Working Paper 17043.

Ductor, L. (2015), "Does Co-authorship Lead to Higher Academic Productivity?". *Oxford Bulletin of Economics and Statistics", 77: 385-407. doi:[10.1111/obes.12070](https://doi.org/10.1111/obes.12070)

Einav, L. y Yariv, L. (2006). "What's in a surname? The effects of surname initials on academic success." *Journal of Economic Perspectives*, 20(1):175–187.

Ellison, G. (2002). "The Slowdown of the Economics Publishing Process", *Journal of Political Economy*, 110:5, 947-993

Hamermesh, D. S. (2013). "Six Decades of Top Economics Publishing: Who and How?" *Journal of Economic Literature*, 51 (1): 162-72.

Hamermesh, D. S. (2018). "Citations in economics: Measurement, uses, and impacts." *Journal of Economic Literature*, 56(1):115–56.

Heckman, J. J. y Moktan, S. (2018). "Publishing and Promotion in Economics: The Tyranny of the Top Five", NBER Working Paper 25093, National Bureau of Economic Research.

Hengel, E. (2019). "Gender differences in citations at top economics journals. Even more evidence that women are held to higher standards in peer review". *Mimeo*, University of Liverpool.

Laband, D. N. y Piette, M. J. (1994). "Favoritism versus Search for Good Papers: Empirical Evidence Regarding the Behavior of Journal Editors", *Journal of Political Economy*, 102:1, 194-203

Lindsey, D. (1989). "Using citation counts as a measure of quality in science measuring what's measurable rather than what's valid." *Scientometrics*, 15(3-4):189–203.

Oswald, A. J. (2007). "An Examination of the Reliability of Prestigious Scholarly Journals: Evidence and Implications for Decision-Makers", *Economica*, 74, 21-31

Price, de Solla J. D.(1978). "Editorial Statement". *Scientometrics* Volume 1, Issue 1.

Quandt, R. E. (1976). "Some Quantitative Aspects of the Economics Journal Literature", *Journal of Political Economy*, 84:4, Part 1, 741-755.

Van Praag, C. M. y van Praag, B. (2008). "The benefits of being economics professor A (rather than Z)." *Economica*, 75(300):782–796.

Wood, D. A. (2016). "Comparing the Publication Process in Accounting, Economics, Finance, Management, Marketing, Psychology, and the Natural Sciences." *Accounting Horizons*, Vol. 30, No. 3, pp. 341-361.

Yan, E. y Ding, Y. (2012). "Scholarly network similarities: How bibliographic coupling networks, citation networks, cocitation networks, topical networks, coauthorship networks, and coword networks relate to each other." *Journal of the American Society for Information Science and Technology*, 63(7):1313–1326.

Zimmermann, C. (2012). "Academic Rankings with RePEc". *FRB of St. Louis Working Paper No. 2012-023A*, Federal Reserve Bank of Saint Louis.