UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

PONIENDO LAS CIENCIAS DE LA COMPUTACIÓN EN EL MAPA: DESARROLLANDO UN SISTEMA PARA GEOLOCALIZAR INVESTIGACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

FELIPE NICOLÁS MANEN NÚÑEZ

PROFESOR GUÍA:
AIDAN HOGAN

MIEMBROS DE LA COMISIÓN:
FELIPE BRAVO MARQUEZ
DANIEL PEROVICH GEROSA

SANTIAGO DE CHILE
2021

Resumen

En el presente trabajo se muestra el desarrollo de un sistema que permite geolocalizar artículos de ciencias de la computación. Este sistema se pensó desde la ausencia de sistemas en la web que tengan la característica de agrupar publicaciones científicas con respecto al lugar de origen o trabajo de sus autores. Se cree que un sistema con esta cualidad podría fomentar la colaboración entre expertos de un área, mejorar la organización de conferencias y ayudar a investigadores en la elección de lugares para estudiar o trabajar. Para lograr esto, se propusieron los objetivos específicos de obtener datos adecuados para resolver el problema, crear métodos para agrupar publicaciones según su geolocalización o por sus temas de investigación y, finalmente, desarrollar interfaces amigables para usuarios.

Para la realización de este trabajo, en primer lugar, se hizo una revisión de distintas herramientas y sistemas que tenían relación con el tema. Entre éstas se destacan fuentes de datos como DBLP, Open Academic Graph y Semantic Scholar. También se vieron algunas alternativas que podrían ser de utilidad para geolocalizar información como herramientas de *entity linking* y Wikidata. A su vez se presentan alternativas para visualizaciones como OpenStreetMap y Leaflet.

Luego, se presentan las tres partes principales del sistema. La primera consistió en una preparación de los datos. Se utilizó la fuente de DBLP, que provee información sobre publicaciones, conferencias y autores de ciencias de la computación. A los datos correspondientes a publicaciones se les asignó una afiliación mediante un algoritmo y luego, utilizando la herramienta de *entity linking* OpenTapioca en conjunto con Wikidata, se logró hacer la geolocalización de estas afiliaciones. La segunda parte del trabajo tenía relación con indexación de estos datos en el motor de búsqueda utilizado Elasticsearch, el cual cumpliría con la función de realizar búsquedas de manera eficiente sobre temas de investigación del área. Como tercera parte se muestra el desarrollo del *frontend* y la experiencia de usuario. Para esto se utilizó el *framework* Django vinculado con Elasticsearch. La visualización principal del sistema consistió en un mapa, creado con la librería Leaflet, que entrega afiliaciones en todo el mundo que hayan participado en la escritura de un artículo sobre un tema inicialmente consultado.

Una vez que el sistema estuvo desarrollado, se muestra la evaluación de todas las partes importantes de él. Se evalúa la precisión del algoritmo para asignar afiliaciones, se compara el rendimiento de distintas herramientas de *entity linking*, se ve el desempeño de los tiempos de respuesta del sistema y se muestran los resultados de un cuestionario hecho a usuarios para ver su percepción. Con esta evaluación se pudo finalmente concluir sobre los objetivos inicialmente planteados. Se cree que el objetivo de desarrollar un sistema con las características propuestas se cumplió. Sin embargo, algunos objetivos específicos no fueron satisfechos en su totalidad.

Agradecimientos

Quiero darme el espacio para agradecer a personas que estuvieron presentes tanto en el proceso de esta memoria como durante toda mi trayectoria en la universidad.

Agradezco a Norma y Andrés, mis padres, quienes siempre se esforzaron por darme una vida plena. Me dieron la oportunidad de estudiar y de que nunca me faltara nada. Por esto estaré eternamente agradecido.

Agradezco a Daniela, mi compañera, quien ha sido mi principal apoyo en todo mi paso por la universidad (y también en el desarrollo de este trabajo). Siempre ha estado alentándome y teniendo fe en mí, aun cuando ni yo mismo la tenía. Gracias infinitas por todo.

A mi hermano Andrés le agradezco pues he vivido con él todos estos años en Santiago, tiempo en el que mutuamente nos hemos apoyado en las situaciones buenas y las no tanto.

A mis amig@s de Puerto Montt, con quienes compartí la experiencia de cambio de ciudad. Han sido un apoyo hasta el día de hoy y son mi segunda familia. Agradezco a Loreto, Carolina, Mauricio y Fernando.

A mis amigos de la universidad. Han sido varios años de apoyo mutuo, no sólo en lo académico sino también en lo humano. Agradezco a Benja, Luis, Nacho y Gus. De manera especial, agradezco a Alfredo y Danny quienes fueron mis primeros amigos en Beauchef y siempre han estado presentes.

A mis amig@s de psicología. Les agradezco porque me apoyaron en todas las decisiones que tomé y, pese a todo, seguimos siendo tan cercanos como en los primeros años. Agradezco a Rocío, Vale y Pablo.

A Aidan, mi profe guía. Le agradezco el darme tranquilidad y herramientas durante el proceso de este trabajo de título. También por orientarme cuando estaba tomando decisiones importantes en mi vida. Ha sido un gran apoyo y ejemplo tanto en lo académico como en lo humano.

Finalmente, agradezco a Hernán, Daniel y Henry por orientarme en aspectos técnicos relacionados a este trabajo. También a Angélica por orientarme cuando lo necesité. Y a Janet por calmar mi ansiedad en el año 2015.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Índice de Código Fuente

Introducción

Actualmente, la ciencia está en constante desarrollo. Día a día investigadores en todo el mundo están probando nuevas técnicas en sus disciplinas y haciendo descubrimientos para la sociedad. Por esto, las búsquedas en la web sobre investigaciones se tornan muy importantes, pues son una manera fácil y directa de saber qué es lo que la comunidad científica está realizando o ha realizado en el pasado. A su vez, las búsquedas son muy útiles tanto para una persona que simplemente tiene deseo de aprender más sobre un tema, como para otra persona que trabaja en ciencia que puede necesitar colaboración en su área de investigación o inspiración para abordar un problema. Lamentablemente, en la actualidad, las herramientas disponibles para acceder a información si bien son capaces de agrupar estos datos según algunos filtros o requisitos que el usuario determina, hay un aspecto que no se logra apreciar en los buscadores disponibles. Esta característica es la agrupación de datos según el lugar de procedencia (país, región, continente, etc.) o, como se tratará desde ahora, la geolocalización del trabajo en ciencia.

La geolocalización del trabajo en ciencia es un aspecto difícil de hallar en los datos que los distintos buscadores entregan. Si bien es posible, por ejemplo, encontrar el país en que un autor trabaja o la ciudad de realización de una conferencia, esta agrupación actualmente no se puede hacer en la otra dirección, es decir, la información no se ha podido organizar según el país, la ciudad o la región en que se realiza. Si una persona quiere buscar todas las conferencias, las publicaciones o los temas de investigación relacionados a un país o región, no lo puede realizar de manera directa, teniendo siempre que buscar dicha información de forma autónoma en la descripción de la revista o artículo, para luego recopilarla y analizarla. En un escenario ideal lo más práctico y eficiente, desde el punto de vista del usuario, sería tener la posibilidad de encontrar esta data de manera automática. De poder cumplirse esto, la comunidad científica se vería beneficiada pues se fomentaría de una manera fácil la colaboración entre investigadores de una región en específico, se facilitaría la labor interdisciplinaria entre expertos de distintos temas en una zona, mejoraría la elección de lugares para la realización de conferencias o postular a posiciones académicas podrían basarse en la densidad de investigación sobre un tema en un sitio.

Con esto como contexto, se propone desarrollar un sistema que permita visualizar datos relacionados con la ciencia (autores, publicaciones, conferencias, etc.) organizados por región (país, continente, zona) de origen o realización. Se propone trabajar con datos de ciencias de la computación pues es un área de investigación muy importante en la actualidad. Además, se tienen muchos datos disponibles para trabajar y se cree que los investigadores de esta área de investigación tendrían una mejor disposición al momento de utilizar una plataforma con

estas características por su afinidad por los sistemas de información. Es importante destacar que se entiende este primer acercamiento como un paso importante para, en el futuro, poder abarcar más disciplinas.

Objetivos

Objetivo General

El objetivo general de este trabajo de memoria es desarrollar un sistema que permita realizar búsquedas sobre investigación en ciencias de la computación, agrupando los resultados según el lugar de producción y visualizando éstos. El caso interesante de este sistema es poder visualizar de buena manera cómo es la producción de literatura científica según afiliación, ciudad o país de origen.

Objetivos Específicos

Los objetivos específicos para poder cumplir el objetivo general son los siguientes:

1. Obtener datos relacionados con ciencias de la computación. Estos datos deben incluir publicaciones, autores, conferencias y revistas del área.
2. Desarrollar un método para organizar datos de artículos y publicaciones según su geolocalización. Para validar este método, se revisará de forma manual una muestra significativa de lo organizado y se chequeará que esté correcta.
3. Desarrollar un método para organizar datos de artículos y publicaciones según el tema de investigación. Para validar este método, se revisará de forma manual una muestra significativa de las publicaciones categorizadas y se chequeará que esté correcta.
4. Desarrollar visualizaciones amigables para los usuarios. Esto se validará haciendo pruebas de usuario, recopilando, luego, la apreciación del sistema.

Dados estos, en el presente informe se presenta el trabajo realizado para poder lograrlos. En el Capítulo 1 se presenta el estado del arte asociado al trabajo, donde se profundizará en distintas herramientas e iniciativas que tienen relación con él. En el Capítulo 2 se muestra la arquitectura del trabajo, una mirada general de lo realizado. En el Capítulo 3 se puede apreciar el tratamiento de datos sobre literatura de ciencias de la computación y el uso de distintas herramientas para poder geolocalizar esta investigación. En el Capítulo 4 se muestra el esquema de base de datos utilizado en el sistema que se desarrolló. El Capítulo 5 se centra en el aspecto del *frontend* del sistema y la experiencia de usuario. El Capítulo 6 corresponde a la evaluación del trabajo y, finalmente, se encuentra la conclusión.

Capítulo 1

Estado del Arte

En este capítulo se enunciarán distintas herramientas, iniciativas y conceptos que se creen son muy importantes para entender y dar contexto al trabajo realizado. En él se mostrarán 7 tópicos generales, los cuales se desglosarán, salvo en el primer caso, en herramientas particulares.

1.1. Wikidata

Wikidata [21] es una base de conocimiento que, tal cual se enuncia en su descripción, puede ser editada por humanos y máquinas. La idea es que sea una base de datos disponible en la web para consultar todo tipo de información. Los datos pueden ser añadidos voluntariamente por usuarios, así como editados y eliminados. Es una iniciativa muy importante dentro del área de investigación correspondiente a la Web Semántica. Dentro del contexto del trabajo, se cree que es una herramienta de mucha utilidad porque en ella se puede encontrar información relevante de la mayoría (sino todos) los países y localidades más importantes del mundo, lo cual ayudaría mucho a la hora de poder hacer geolocalización de la investigación. A su vez, cuenta con un servicio para realizar consultas mediante el lenguaje SPARQL. Un ejemplo de los datos que pueden ser obtenidos se puede apreciar en las Figuras 1.1 y 1.2, donde se muestra la entidad Universidad de Chile y se pueden obtener, por ejemplo, el país donde está ubicada y sus coordenadas geográficas exactas.

1.2. Sistemas de información y datos científicos

En este momento, la web ofrece una diversidad de herramientas para encontrar documentos o información sobre la actividad científica en el mundo. Dentro de ellas, hay algunas básicas, mientras que otras se acercan un poco más al problema que se quiere abordar. A continuación, se hará una revisión de los ejemplos que se creen son los más relevantes con respecto al tema a trabajar, mostrando sus principales características.

- Google Scholar: es posiblemente el motor de búsqueda académico/científico más utilizado en la actualidad. Con esta herramienta, un usuario puede encontrar documentos relacionados con el mundo científico, ya sean artículos, revistas, tesis, etc. A su vez,

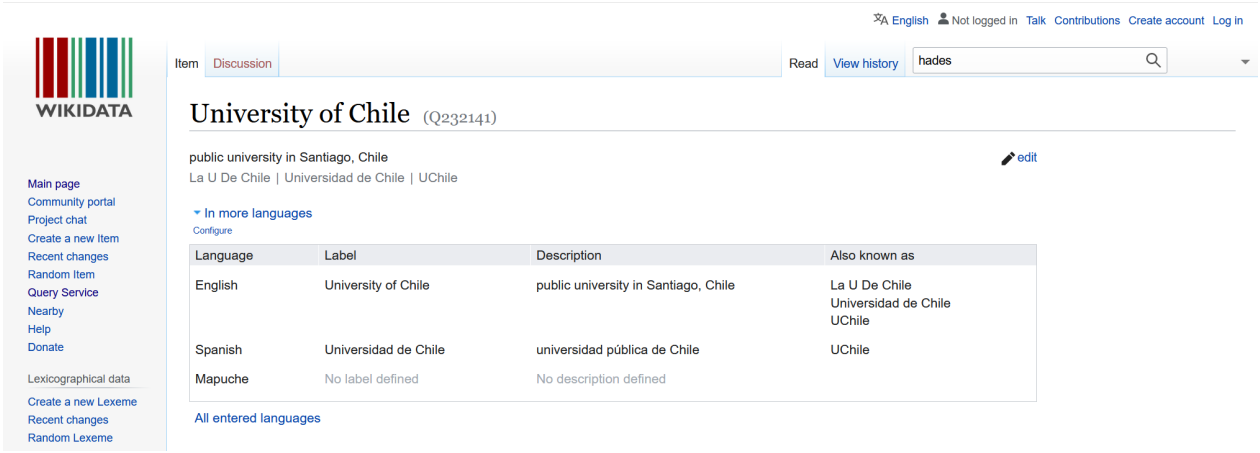

Figura 1.1: Ejemplo de objeto de Wikidata.

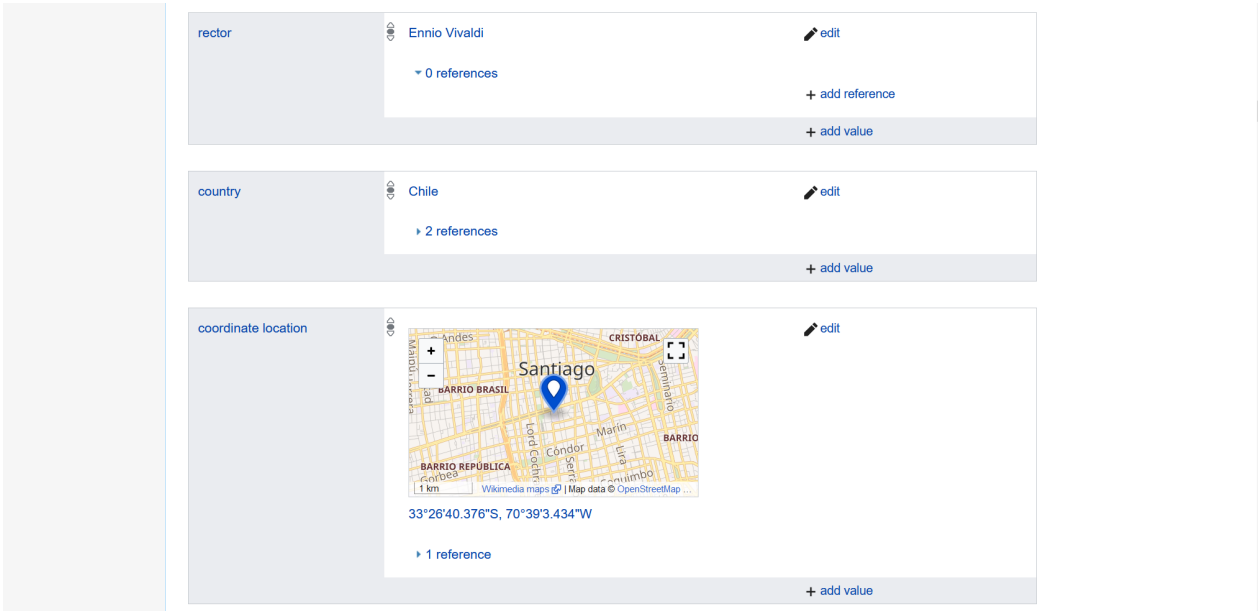

Figura 1.2: Ejemplo de datos entregados por Wikidata.

este motor posee la característica de un perfil de autor, donde se resumen tanto las producciones en las que ha participado, sus coautores, las citas en otros trabajos, resumen estadístico de esta información, entre otros aspectos relevantes. Este perfil soporta una afiliación del autor, la cual, en general, suele ser la universidad o laboratorio donde realiza sus labores académicas.

- DBLP [11]: es un sitio web que centraliza información bibliográfica sobre publicaciones o eventos relacionados con las ciencias de la computación. En él, se puede explorar a través de las siguientes categorías: los nombres de los autores, conferencias relacionadas con el área (indicando dentro de su información el lugar de realización de ella), revistas y series. Adicionalmente, el sitio soporta búsquedas, que arrojan sus resultados organizados según dichas categorías. Como aspecto secundario, el sitio incorpora distintos datos estadísticos según distintas clasificaciones, como el tipo de publicación, publicaciones por año, número de autores por publicación, entre otros.

- ResearchGate: es una red social científica. En ella, un usuario puede realizar búsquedas sobre otros autores o publicaciones, explorar a través de otros escritos que el sitio recomienda, ver los coautores asociados o los temas de interés asociados a un autor, etc.
- Pubmed: es el principal motor de búsqueda de referencias del área médica. Su característica más llamativa es que permite, a través de operaciones booleanas, realizar búsquedas avanzadas que permitan encontrar publicaciones del área según autores, afiliaciones, lenguaje en que fue escrito y muchas otras características que el usuario puede definir.
- ArnetMiner [20]: es un sistema que permite, a partir de datos sobre publicaciones académicas en la web, generar, con minería de datos, asociaciones entre autores, publicaciones y conferencias. Esta herramienta se ha utilizado para crear perfiles de usuario, hacer ranking de conferencias, estudiar alternativas de *expert finding*, etc. A su vez, un aspecto interesante de este sistema, es que permite hacer una simulación en un mapa de todos los lugares en el mundo en que un autor ha participado, según su trayectoria de publicaciones científicas.
- Open Academic Graph (OAG): es un grafo de conocimiento que enlaza los correspondientes a Microsoft Academic Graph [19] y el grafo de conocimiento de ArnetMiner. Contiene datos de 208.915.369 artículos científicos obtenidos del primer grafo, 172.209.563 obtenidos del segundo y hace 91.137.597 vínculos entre ambos.
- Semantic Scholar [1]: es una herramienta asociada a literatura científica. A su vez también provee un conjunto de datos en formato JSON que incluye información de artículos tales como autores, citas, año de publicación, entre otras. La información puede obtenerse a través de su respectiva API o a través de su corpus directamente.

Es importante destacar que si bien todos los ejemplos mencionados son muy útiles a la hora de consultar y obtener información relevante sobre tópicos de ciencia, autores de publicaciones, datos sobre conferencias, hasta ciertas conexiones entre investigadores con sus investigaciones, ninguno de ellos tiene la capacidad de hacer agrupaciones o filtros según una región geográfica del planeta. Hoy en día, no existe un sistema que, por ejemplo, dado un país, muestre todos los investigadores de un área en él, o que dado una región en particular, entregue un resumen de los tópicos de investigación más relevantes.

1.3. Herramientas de *entity linking*

Dada la naturaleza del problema que se quiere abordar, se requiere poder hacer una localización fina de los lugares en que se hace investigación de ciencias de la computación (ubicar por ejemplo, en qué ciudad se encuentra la Universidad de Harvard). Dados los datos disponibles, esta información no está directamente, es decir, es fácil encontrar afiliaciones a instituciones o universidades pero no necesariamente el país o región en que se encuentran ellas. En este sentido, el área del procesamiento de lenguaje natural *entity linking* [22] permite hacer este tipo de asociaciones, y encontrar de una manera eficiente la geolocalización. Un ejemplo del proceso de *entity linking* es el siguiente. Si se ingresa el texto "Gabriela Mistral was born in Vicuña, Chile.", se espera que la herramienta resalte las entidades "Gabriela Mistral" y "Vicuña, Chile". En este aspecto existen algunas herramientas que cumplen con aquel objetivo, todas disponibles como APIs:

- DBpedia Spotlight [12]: es una herramienta que permite hacer vínculo entre entidades de un texto y secciones de DBpedia [10] (proyecto que extrae información estructurada de Wikipedia, generando una base de datos). La herramienta permite aplicar los procesos de *entity extraction*, *entity recognition* y *name resolution*. En línea se provee una demostración de esta herramienta, donde un usuario puede insertar un texto y se identifican las entidades de él, haciendo el link a su sitio en DBPedia.
- Babelfy [14]: es una herramienta que permite hacer desambiguación de texto y también detección de entidades como en el caso anterior. Lo interesante de Babelfy es que permite hacer estos procesos sobre un gran número de idiomas.
- TAGME [7]: es otra herramienta de identificación de entidades, que promete ser muy veloz y efectiva para frases cortas, haciendo el vínculo directo al sitio de Wikipedia de la entidad.
- Aida [23]: la idea de esta herramienta es la misma que las anteriores, salvo que el vínculo se hace a la base de conocimiento YAGO2 [9].
- OpenTapioca [5]: esta es una herramienta muy nueva en comparación con las anteriormente mencionadas, que permite hacer vínculos con la base de conocimientos Wikidata [21].

1.4. Herramientas de Geolocalización de Texto

Un aspecto que podría ser de mucha utilidad para el trabajo que se quiere realizar son las herramientas que se enfocan directamente en poder identificar el lugar geográfico de un texto o la geolocalización del texto. Investigadores del área de procesamiento del lenguaje natural han hecho avances en este tema y han desarrollado algunos sistemas que permitan realizar esta acción. A continuación se mencionan algunos de ellos:

- *pigeo* [15]: es una librería del lenguaje de programación Python para predecir ubicación geográfica, dado un texto o el un usuario de la red social Twitter. Está basado en modelos pre-entrenados y permite a desarrolladores crear sus propios modelos predictivos.
- CLIFF-CLAVIN [6]: es una herramienta que permite analizar texto de noticias, identificando lugares, organizaciones y personas mencionadas en ellas. Con este proceso, luego, se logra hacer un análisis fino, y poder geolocalizar el texto, identificando su localización de manera específica.

1.5. Herramientas de Visualización Geográfica

Actualmente, el universo de librerías de *frontend* está en crecimiento y hay muchas opciones a escoger para poder abordar el problema de la geolocalización en ciencia. A continuación se presentan 3 herramientas que son importantes de destacar:

- Vue.js: es un *framework* de JavaScript que permite hacer visualizaciones en base a componentes. Lo interesante de Vue es que es de código abierto y muchas personas contribuyen en él. Adicionalmente, Vue cuenta con algunas componentes dedicadas a la visualización de mapas. Hay una en particular que resulta ser la más interesante, Choropleth Map [18], pues permite subdividir el territorio en regiones y a su vez,

permite medir la densidad de alguna característica (ver Figura 1.3). En este caso sería de mucha utilidad si se quiere mostrar el número de publicaciones o autores asociados a una región en específico.

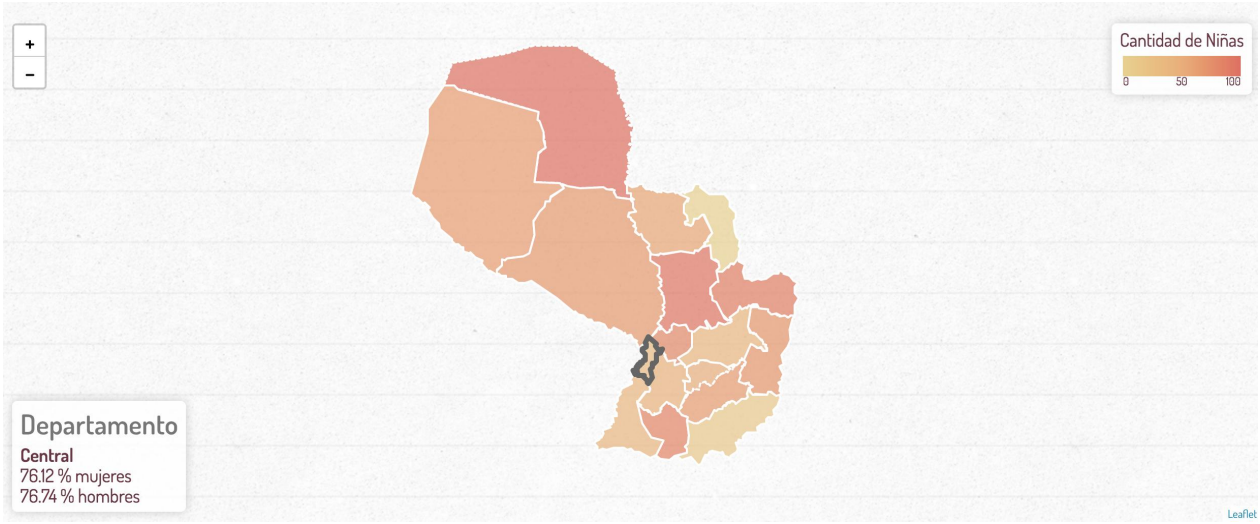

Figura 1.3: Ejemplo de componente Choropleth Map, extraído del repositorio del proyecto [18]. En él se muestra un mapa de Paraguay, con todos sus departamentos delimitados. El color indica la densidad de niñas que asisten a escuelas del respectivo departamento.

- OpenStreetMap [8]: es un proyecto para crear mapas editables. La iniciativa nace de la necesidad de tener información geográfica de libre uso y es muy usada en la actualidad. El proyecto es muy versátil y es compatible con muchos frameworks conocidos para el desarrollo web.
- Leaflet: es una librería del lenguaje de programación Javascript, que permite realizar mapas interactivos de manera simple. Entre sus características más importantes están que permite ubicar marcadores dentro de mapas, agruparlos en conjuntos en caso de que hayan muchos en un sector del mapa y además, la creación de *pop-ups* asociados a estos marcadores. Un ejemplo de su uso se puede apreciar en la Figura 1.4.

1.6. Bases de datos NoSQL

Si bien las bases de datos relacionales son quizás las más utilizadas hoy en día para sistemas de información, es importante destacar que hay algunas bases de datos diferentes, usualmente asociadas como NoSQL, que han demostrado tener un mejor desempeño a la hora de hacer búsquedas y entregar información consultada. A continuación se presentan dos sistemas de bases de datos muy utilizadas en la actualidad, que están basadas en la indexación de documentos.

- MongoDB [4]: se basa en la utilización de documentos de formato BSON. Permite indexar documentos de manera flexible, es decir, dos documentos no necesariamente deben tener los mismos campos. En el caso de MongoDB, se ha visto un buen desempeño en tiempos de inserción, actualización y eliminación de un gran volumen de datos en comparación con bases de datos relacionales [3].

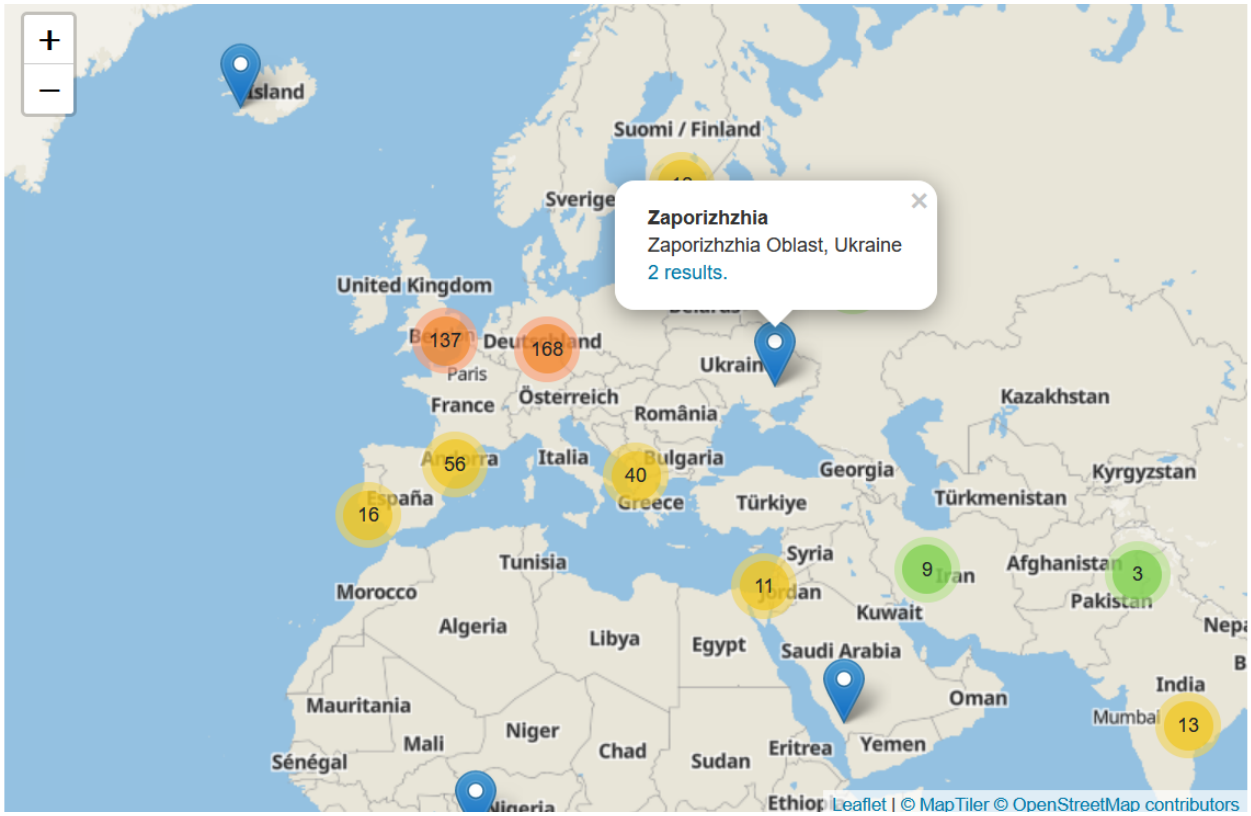

Figura 1.4: Ejemplo de mapa generado con Leaflet. Se aprecian marcadores, conjuntos y *pop-up* asociado a uno de ellos.

- Elasticsearch: se basa en la utilización de documentos de formato JSON y está basado en Apache Lucene [2], una API de código abierto orientada a la recuperación de información. En comparación con MongoDB, Elasticsearch permite hacer búsquedas de texto muy eficientes. A su vez permite configurar un índice de documentos con distintas opciones, incluyendo operaciones conocidas de procesamiento de lenguaje natural como por ejemplo stemming o tokenización, para de esta manera propiciar un buen rendimiento para consultas de interés.

1.7. *Topic modelling*

Un área importante en la minería de texto es el modelamiento de temas o *topic modelling* de un texto. La idea es poder extraer temas en base a un texto en particular, es decir, si hay más palabras relacionadas con un tema en particular, se encasillará el texto en aquel tema. En el caso del trabajo, son de interés herramientas que puedan encontrar temas académicos o de investigación, pues dado el objetivo, se espera poder agrupar publicaciones según el tema específico del cual tratan. A continuación se enuncia una herramienta que se cree tiene mucha relación con el trabajo.

- CSO Classifier [16]: es una herramienta de clasificación de texto según temas. Utiliza la Computer Science Ontology [17] (CSO) una ontología creada sobre 16 millones de artículos relacionados con ciencias de la computación. El clasificador permite categorizar

el texto tanto en áreas de investigación generales como específicas.

Capítulo 2

Arquitectura de la solución

En este capítulo se presentará una vista general del sistema desarrollado. Esta vista general se hará mostrando los aspectos fundamentales del sistema. A su vez se dará una corta justificación de la elección de cada una de las tecnologías que se usaron para construir el sistema. Además de todo esto, se mostrarán algunas dificultades que se tuvieron que manejar a lo largo del trabajo.

En una mirada global, el sistema que se desarrolló consistió en una aplicación web que permitiera hacer consultas sobre algún tema relacionado a las ciencias de la computación y que, como resultado, entregara un mapa señalando todos los lugares en el planeta donde se desarrollara investigación en aquel tópico. ¿Qué se entiende por "desarrollar investigación"? En el trabajo la definición se simplifica a la participación de un investigador, asociado a una institución, en la escritura de un artículo científico relacionado a aquel tema inicialmente consultado. La idea del sistema es poder generar una vista general de cómo se da la investigación de los distintos temas de las ciencias de la computación en todo el mundo, y que de esta manera pueda ser una herramienta para ser usada por científicos del área. Los principales casos de uso del sistema son generar colaboración entre expertos de un área, buscar locaciones para continuar estudios, escoger lugares para la realización de conferencias, entre otros. Teniendo esta definición del propósito del sistema, es importante entonces detallar cuáles son las distintas partes que lo componen. A continuación una aproximación en particular de cada una de ellas.

2.1. Datos, geolocalización y temas de investigación

Los datos utilizados en el sistema fueron los que provee DBLP en su página web. En ellos se pueden hallar datos relacionados con artículos científicos, publicados en revistas científicas o en conferencias, nombre de las conferencias realizadas a través de los años, páginas web de autores y algunas tesis de maestría y doctorado. En lo que concierne al proyecto, se utilizaron los datos de cerca de 5.000.000 de artículos científicos. Estos datos fueron bastante completos para dar la información principal de un artículo científico: título, año de publicación, conferencia o revista donde fue presentado, autores, Digital Object Identifier (DOI), etc. Sin embargo, éstos carecían de información relevante sobre **afiliaciones de los autores** al mo-

mento de su publicación (información que usualmente se puede hallar en los encabezados de los artículos) así como de **los temas de investigación** asociados al artículo. En este sentido, hubo que tomar ciertas decisiones para, aun así, poder converger a un sistema que pudiera ser usable, tuviera las características antes mencionadas y entregara resultados razonables.

Para poder resolver el problema de la poca información sobre afiliaciones de los autores de un artículo de investigación, se consideraron dos alternativas. La primera de ellas fue utilizar Open Academic Graph como fuente de datos, pues para algunos artículos proveía la información que usualmente se encuentra en los encabezados con respecto al lugar de trabajo de los autores. Sin embargo, esta idea se desechó pues, tras hacer una exploración de aquellos datos, no era sencillo hacer el cruce de información entre esta fuente con la de DBLP y, por otro lado, tampoco proveía la información de las afiliaciones para una gran cantidad de artículos. La segunda alternativa, que fue la que finalmente se escogió, consistió en un tratamiento de los datos de DBLP. Luego de un estudio más en profundidad sobre estos datos, se logró observar que, de manera indirecta, éstos proveían información de afiliaciones, principalmente a través de los datos correspondientes a las tesis doctorales y las páginas web de autores. Así, a través del diseño de un algoritmo, se logró asignar una afiliación a una gran cantidad de los artículos (cerca del 50 % de ellos) considerando las afiliaciones de sus autores. Una vez se asignó una afiliación a estos artículos, se procedió a hacer su respectiva geolocalización, es decir poder asignar el lugar geográfico exacto de aquella afiliación. El proceso fue bastante directo: se utilizó una herramienta de *entity linking* para identificar la afiliación como una entidad y, luego identificada, se hizo una consulta sobre sus datos geográficos usando el lenguaje SPARQL sobre Wikidata, obteniendo de esta manera la ciudad, el país y las coordenadas de la afiliación. Más sobre todo el tratamiento de los datos, el uso de *entity linking* y la geolocalización se puede encontrar en el Capítulo 3.

Por otro lado, para la dificultad con respecto a los temas de investigación, también se consideraron varias alternativas. En primer lugar, se consideró también usar Open Academic Graph pues, para algunos artículos, esta fuente de datos proveía palabras clave (en inglés *keywords*), información usualmente disponible en el encabezado de los escritos científicos. Se cree que estas palabras clave, al ser escogidas por sus mismos autores, son un fiel reflejo de los temas generales de cada artículo. Lamentablemente, al igual que en el caso anterior, OAG no proveía estos datos para una gran cantidad de publicaciones, por lo que esta idea se terminó desestimando. Como segunda alternativa, se consideró usar la API de Semantic Scholar, pues dentro de los resultados que entrega al consultar por un artículo también provee las *keywords* del mismo. Sin embargo, la tasa de uso de la API es muy limitada (100 peticiones en una ventana de 5 minutos por dirección de IP), por lo que, si bien los datos eran bastante completos en comparación con OAG, no era factible utilizar esta herramienta pues se debía obtener las palabras clave de un volumen alto de artículos, lo cual no era compatible con esta limitación. Como tercera alternativa, se intentó utilizar CSO classifier [16], pero también se debió desistir pues el tiempo que tomaba en clasificar un texto era cercano a 3 segundos y, nuevamente, se tornaba incompatible con los plazos relacionados al proyecto, dado, nuevamente, el alto volumen de artículos que se necesitaba clasificar (cercano a 5.000.000). Finalmente, se tomó la decisión de simplemente hacer la búsqueda de texto de temas en los títulos y nombres de las conferencias o revistas donde los artículos eran publicados. Esta búsqueda de texto se logró hacer de manera eficiente y, si bien, se entiende que los temas no siempre están explícitos en los títulos del artículo, de las conferencias o revistas, se cree que se obtienen resultados

razonables en esta exploración inicial relacionado al tema del trabajo. Más sobre la búsqueda de texto en la siguiente sección y en el Capítulo 4.

2.2. Esquema de base de datos

Para la base de datos se pensó desde un principio en utilizar una alternativa NoSQL. Esto debido a la escalabilidad de los datos y también a que, dada la naturaleza de los datos de DBLP, no se tendría un esquema claro: se necesitaba una alternativa lo más flexible posible. A su vez, dada la decisión mencionada antes, de realizar la búsqueda de tópicos simplemente como una búsqueda de texto en los títulos de cada artículo y la conferencia o revista donde se presenta, se decidió utilizar Elasticsearch como motor de búsqueda. Elasticsearch provee flexibilidad al momento de inserción y modificación de datos, ya que no requiere de un esquema preestablecido para funcionar, como sí es el caso de un modelo relacional. Además, como se mencionó en el Capítulo 1, permite configurar un índice (similar a una base de datos en un modelo relacional) para poder realizar consultas más precisas o eficientes sobre los campos de interés. En el caso del proyecto, el índice almacena documentos sobre los artículos (un documento corresponde a un artículo) y se puso mayor énfasis en los campos relacionados con el título, la revista de publicación o la conferencia de presentación. Más sobre esto en el Capítulo 4.

2.3. *Frontend* y experiencia de usuario

Una vez que el modelo de datos estaba establecido, se procedió a trabajar en la arista del *Frontend* y la comunicación del sistema con el usuario. Para esto se decidió trabajar con el *framework* Django del lenguaje de programación Python, que permite crear aplicaciones web separando su lógica en tres. Se decidió usar este *framework* por la familiaridad, pues el estudiante ya había desarrollado muchos proyectos usándolo. A su vez, se cree que es una herramienta simple de usar y permite desarrollar aplicaciones web con funcionalidades complejas. En particular, el *framework* utiliza el patrón de diseño *Model-Template-View* (Modelo-Plantilla-Vista o MTV), que es un patrón de diseño similar a *Model-View-Controller* (Modelo-Vista-Controlador o MVC). Para el caso del trabajo, el Modelo corresponde al modelo de datos diseñado en Elasticsearch, explicado en el apartado anterior, que es la parte de la arquitectura que almacena y provee los datos consultados o solicitados. La Vista, a diferencia de una arquitectura MVC, se encarga de ejecutar la lógica de negocio, comunicarse con el Modelo y luego renderizar una plantilla. La Plantilla muestra lo que el usuario ve, utilizando los datos y la información que la Vista obtuvo del Modelo. En el caso de MTV, a diferencia de MVC, no existe una lógica de Controlador por separado. Para las Plantillas, el framework usa por defecto HTML para el contenido estático y el lenguaje Javascript para las respuestas que el sistema debe dar al usuario. Además se decidió utilizar la librería Bootstrap, para poder usar fácilmente plantillas preexistentes, componentes más llamativas y así poder diseñar e implementar interfaces gráficas más amigables visualmente.

2.4. Resumen de la Arquitectura

Teniendo claras cada una de las componentes del sistema, en la Figura 2.1 se muestra un diagrama que sintetiza la arquitectura. Se señalan las partes antes explicadas y se explicita a qué sección corresponde el cliente y a cuál corresponde al servidor.

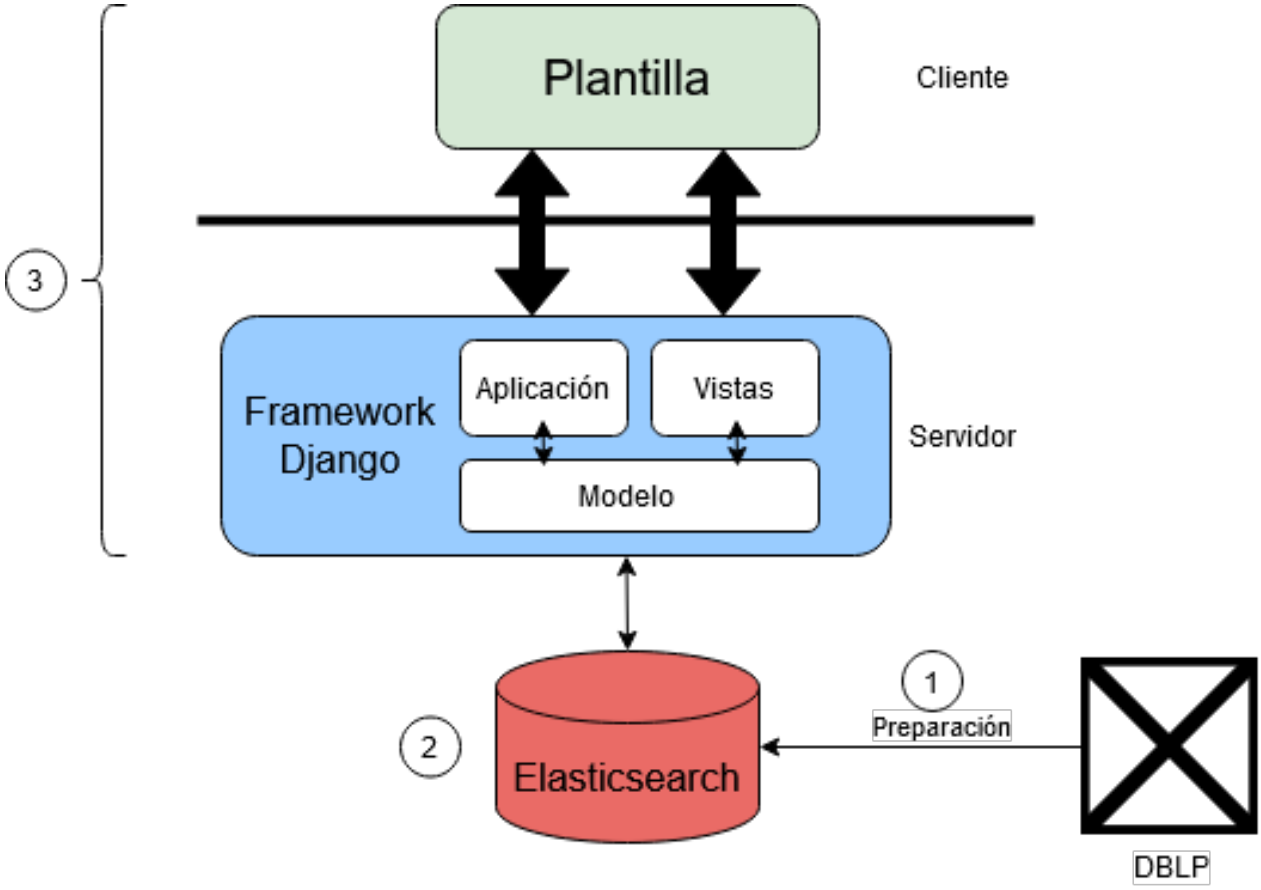

Figura 2.1: Esquema resumen del sistema desarrollado.

Con esta mirada general, en el resto del informe se podrá ver cada una de estas partes de manera más detallada. Los siguientes tres capítulos corresponden a los tres aspectos señalados, respectivamente.

Capítulo 3

Datos, *entity linking* y geolocalización

En este capítulo se describirá todo el trabajo relacionado al tratamiento de los datos. En particular, se describirán los datos de DBLP sobre los cuales se trabajó, cómo se generó la asociación de artículos con alguna afiliación y cómo se lograron obtener los datos geográficos de las distintas afiliaciones encontradas.

3.1. Datos DBLP

Los datos utilizados durante el trabajo, como se mencionó anteriormente, corresponden a un conjunto de datos que provee el sitio DBLP. Estos datos están disponibles en un archivo en formato XML que se va lanzando periódicamente una vez haya un número importante de actualizaciones relacionadas con la publicación de nuevos artículos o la realización de nuevas conferencias. El archivo que se usó durante el trabajo corresponde a una versión del conjunto de datos disponibilizada en julio de 2020. Este archivo se divide en distintos elementos (unidades de un archivo XML) que están categorizados en 8 etiquetas. Es importante destacar que dos elementos de una misma etiqueta no necesariamente tienen los mismos campos de información. En la Tabla 3.1 se detallan las distintas categorías presentes en el archivo, las etiquetas y la cantidad de elementos por etiqueta.

Categoría	**Etiqueta**	**Cantidad de elementos**
Artículos de revista científica	`article`	2.324.643
Artículos de conferencia o workshop	`inproceedings`	2.652.097
Colecciones de conferencia o workshop	`proceedings`	44.783
Tesis doctorales	`phdthesis`	77.119
Tesis de magíster	`mastersthesis`	12
Libros	`book`	18.242
Capítulos de libros o monografías	`incollection`	65.278
Páginas webs	`www`	2.558.760

Tabla 3.1: Distribución de categorías de conjunto de datos de DBLP.

En el contexto del trabajo, los elementos que más interesan son los relacionados a artículos científicos (etiquetas `article` e `inproceedings`), que en total suman 4.976.740, los cuales

poseen todos los datos generales relacionados a una publicación científica: autores, año de publicación, colección de conferencia o revista, según corresponda, en la que fue publicado y algún enlace al artículo, generalmente correspondiente a su respectivo DOI. Para el caso de los artículos de conferencia o workshop no se señala de manera completa el nombre de la edición en que se publicó, sino una llave del evento asociado y su año, por lo que si se quiere obtener la información completa se debe entonces buscar el nombre entre los elementos de etiqueta `proceedings`, teniendo como referencia esta llave. Finalmente, también son de interés tanto las tesis de maestría, de doctorado y las páginas web, pues si bien no son directamente elementos que se quieren geolocalizar en el desarrollo del trabajo, son en particular los que poseen información sobre las afiliaciones de los autores asociados. Son de especial importancia las páginas webs pues muchas de ellas corresponden a las personales de los distintos investigadores. Es importante señalar, adicionalmente, que ningún tipo de artículo posee estos datos de afiliaciones ni tampoco datos relacionados con los temas de investigación. Además, se tiene que para todos los artículos de investigación se encontraron un total de 2.2621.558 autores.

Con esta explicación es importante dar algunos ejemplos con el motivo de dar a entender mejor la estructura de los datos. Para el caso de un artículo, se puede apreciar un ejemplo en el Código 3.1.

```
<article mdate="2017-05-23" key="conf/www/CibranVVSJ07">
    <author>Mar&iacute;a Agustina Cibr&aacute;n</author>
    <author>Bart Verheecke</author>
    <author>Wim Vanderperren</author>
    <author>Davy Suv&eacute;e</author>
    <author>Viviane Jonckers</author>
    <title>Aspect-oriented Programming for Dynamic Web Service
    Selection, Integration and Management.</title>
    <pages>211-242</pages>
    <year>2007</year>
    <journal>World Wide Web</journal>
    <number>3</number>
    <ee>https://doi.org/10.1007/s11280-006-0017-2</ee>
    <url>db/journals/www/www10.html#CibranVVSJ07</url>
</article>
```

Código 3.1: Ejemplo de artículo.

Para el caso de una página web, en algunos casos asociados a autores, se tiene la información detallada sobre sus distintas afiliaciones a lo largo de los años. Lamentablemente para el trabajo, esta información completa no está disponible para un gran número de escritores. Un ejemplo de esto se puede apreciar en el Código 3.2. Es interesante destacar que DBLP para el manejo de autores con los mismos nombres exactos, les añade un número para diferenciarlos. En el Código 3.2 se puede apreciar para el autor Hong Xu, que posee a su lado el número 0004 de desambiguación.

```
<www mdate="2019-01-10" key="homepages/01/5265-4">
<author>Hong Xu 0004</author>
    <title>Home Page</title>
    <note label="since 2009" type="affiliation">
        Nanyang Technological University,
        Division of Psychology, Singapore
    </note>
    <note label="2007-2009" type="affiliation">
        Columbia University, New York, NY, USA
    </note>
    <note label="2000-2007" type="affiliation">
        University of Chicago, IL, USA
    </note>
    <url>https://orcid.org/0000-0003-1389-5408</url>
    <url>https://www.wikidata.org/entity/Q41048161</url>
    <url>https://www.researcherid.com/rid/M-5173-2017</url>
    <url>
        https://www.scopus.com/authid/detail.uri?authorId=57193418618
    </url>
</www>
```

Código 3.2: Ejemplo de página web.

Para el caso de una tesis, doctoral o de maestría, también se tiene información sobre la afiliación del estudiante al momento de presentarla, en este caso se tiene un subelemento de etiqueta `school`.

```
<phdthesis mdate="2017-01-09" key="books/daglib/0081586">
    <author>Frank H&ouml;nes</author>
    <title>Analyse von bin&auml;ren Dokumentbildern.</title>
    <pages>1-216</pages>
    <school>Kaiserslautern University of Technology, Germany</school>
    <year>1995</year>
    <ee>http://d-nb.info/945215533</ee>
</phdthesis>
```

Código 3.3: Ejemplo de tesis doctoral.

Por otro lado, para un artículo de conferencia se tiene un ejemplo como el del Código 3.4.

```
<inproceedings mdate="2019-02-06" key="conf/nooj/KassmiMMM18">
    <author>Rafik Kassmi</author>
    <author>Mohammed Mourchid</author>
    <author>Abdelaziz Mouloudi</author>
    <author>Samir Mbarki</author>
    <title>Implementation of Arabic Phonological Rules in NooJ.</title>
    <pages>16-26</pages>
    <year>2018</year>
    <booktitle>NooJ</booktitle>
    <ee>https://doi.org/10.1007/978-3-030-10868-7_2</ee>
    <crossref>conf/nooj/2018</crossref>
    <url>db/conf/nooj/nooj2018.html#KassmiMMM18</url>
</inproceedings>
```

Código 3.4: Ejemplo artículo de conferencia.

En este caso se tiene como llave de referencia `conf/nooj/2018`. Si se busca aquella llave correspondiente a un elemento de etiqueta `proceedings`, se obtiene que aquel artículo es parte del ejemplo en Código 3.5.

```
<proceedings mdate="2019-02-06" key="conf/nooj/2018">
    <editor>Ignazio Mauro Mirto</editor>
    <editor>Mario Monteleone</editor>
    <editor>Max Silberztein</editor>
    <title>
        Formalizing Natural Languages with NooJ 2018
        and Its Natural Language Processing Applications
        - 12th International Conference, NooJ 2018, Palermo,
        Italy, June 20-22, 2018, Revised Selected Papers
    </title>
    <booktitle>NooJ</booktitle>
    <publisher>Springer</publisher>
    <year>2019</year>
    <series href="db/series/ccis/index.html">Communications
    in Computer and Information Science</series>
    <volume>987</volume>
    <isbn>978-3-030-10867-0</isbn>
    <isbn>978-3-030-10868-7</isbn>
    <ee>https://doi.org/10.1007/978-3-030-10868-7</ee>
    <url>db/conf/nooj/nooj2018.html</url>
</proceedings>
```

Código 3.5: Ejemplo de colección de conferencia.

Como se indicó previamente, y como se puede apreciar en los ejemplos 3.1 y 3.4, los artículos de investigación no poseen información relevante sobre su afiliación. Para esto es que se decidió asociar a autores con alguna afiliación, obteniendo esta información de los elementos de las tres etiquetas mencionadas: páginas web, tesis doctorales y de maestría. Para obtener esta

información de forma que se pudiera utilizar mejor, se ejecutó un programa que permitiera asociar autores con todas sus afiliaciones posibles, usando elementos de aquellas tres etiquetas. De esta manera, se pudo asociar al menos una afiliación a 173.225 autores, lo cual es cercano al 6 % de ellos. Si bien este número no es tan alto, se prevé que dentro de este conjunto que se logró asociar a alguna afiliación se encuentran los autores más importantes del área.

3.2. Algoritmo inicial para asignar afiliaciones

Una vez se tenían los datos de las afiliaciones de los autores, se continuó por asociar una publicación científica con una de ellas. Para esto se diseñó un algoritmo muy simple que permitiera asociar una publicación con una única afiliación que se escoge según sea la más probable. El algoritmo sigue los pasos a continuación:

1. Por cada artículo de investigación, se obtienen las afiliaciones de cada uno de sus autores.
2. Todas las afiliaciones de estos autores se enlistan en un arreglo.
3. Del arreglo de todas las afiliaciones se extrae la afiliación con más frecuencia de aparición dentro del arreglo.
4. Se asigna como afiliación del artículo la encontrada del paso 3.

Una vez se ejecutó el algoritmo sobre todos los artículos del conjunto de datos, se pudo asociar una afiliación a 2.389.437 de ellos, cerca del 49 % del total, los cuales a simple vista mostraron resultados positivos. En total se consideraron 37.203 afiliaciones diferentes. En el Capítulo 6 se muestra una evaluación de este algoritmo.

Naturalmente, este algoritmo entrega resultados incompletos porque un artículo, salvo que sea escrito por un único autor, lo más probable es que tenga más de una afiliación asociada. Se quiso seguir este camino con la intención de poder obtener un resultado más representativos de un paper (al menos una afiliación correcta) y no muchos que fueran poco representativos de las afiliaciones (muchas afiliaciones incorrectas con algunas correctas). Sin embargo, otro aspecto a considerar es el modo en que se obtiene la afiliación. Un ejemplo de un mal funcionamiento es si el conjunto de autores posee sólo 2 afiliaciones asociadas, con la misma frecuencia. En ese caso se escogería de manera aleatoria cuál afiliación es la que finalmente se asocia al artículo. Buscando alternativas para poder mejorar esta falencia, se intentó usar la información temporal de las afiliaciones que algunos autores poseían, asociados a sus tesis de doctorado o al detalle en los elementos de página web del archivo. Lamentablemente, tras hacer algunas pruebas, no se obtuvieron resultados que reflejaran un desempeño sustancialmente mejor. Se espera poder hacer un reajuste a este algoritmo para poder asociar afiliaciones de mejor manera en el futuro.

3.3. Uso de herramienta *entity linking* para identificar entidades

Una vez se asociaron los artículos a sus respectivas afiliaciones, se debía proceder a geolocalizar éstas. Para realizar esto, en primer lugar, se utilizó una herramienta de *entity linking*,

para luego poder hacer consultas sobre Wikidata. Teniendo en cuenta las herramientas mencionadas en la Sección 1.3, se consideraron tres como las más adecuadas a utilizar: DBpedia Spotlight, TAGME y OpenTapioca. Las primeras dos fueron las que más llamaron la atención inicialmente pues son muy utilizadas en la actualidad y tienen un código fuente con mucho soporte. A su vez, como se mencionó antes, ambas hacen vínculo a dos fuentes de datos muy completas como lo son DBpedia y Wikipedia. Sin embargo, tras hacer algunas pruebas, se pudo ver que OpenTapioca obtenía mejores resultados al identificar las afiliaciones (más sobre las pruebas en el Capítulo 6). Además, al hacer vínculo directamente con Wikidata facilitaba el proceso que se debía hacer posteriormente (encontrar los datos geográficos), escenario que se dificultaba al usar DBPedia Spotlight o Tagme, ya que éstas entregan la dirección URL de la entidad reconocida, por lo que se debía hacer antes un procesamiento de texto para luego poder realizar de manera correcta la consulta en Wikidata.

Una vez escogida la herramienta a utilizar, se realizó la identificación de las entidades. Esto se hizo con un código simple que iteraba sobre todas las afiliaciones asociadas a algún artículo, haciendo la consulta de cada afiliación a través de la API de OpenTapioca. Ésta, al consultar por un texto, entrega un texto en formato JSON. Los valores de las llaves `id` y `qid` corresponden al identificador de Wikidata de la entidad reconocida (ver ejemplo en 3.6). Durante este proceso se logró hacer la identificación de las 37.203 afiliaciones distintas que se asocian a los artículos. Una evaluación de los resultados obtenidos usando OpenTapioca se puede hallar en el Capítulo 6. Es importante notar que algunas instituciones no fueron reconocidas por la herramienta pues, al ser ésta una de naturaleza probabilística, reconocía otra como la más probable. Por ejemplo, Google France, es reconocida como Google. Esto, luego, se traduciría en una falla en los datos pero que, viéndolo de manera global, no es muy significativo.


```
{
    "text": "University of Chile",
    "annotations": [
        {
            "start": 0, "end": 19,
            "tags": [
                {"id": "Q232141",
                "label": ["University of Chile"],
                "aliases": [...],
                "extra_aliases": null,
                "desc": "public university in Santiago, Chile",
                "nb_statements": 83,
                "nb_sitelinks": 53,
                "edges": [...],
                "types": {...},
                "rank": 10.795763850882299,
                "score": 0.5683612915715982,
                "valid": null}
            ],
            "best_qid": "Q232141",
            "log_likelihood": 22.49248849823838
        }
```

```
    ]
}
```

Código 3.6: Ejemplo de resultado de API de OpenTapioca para el texto University of Chile.

3.4. Geolocalización usando SPARQL y Wikidata

Finalmente, una vez ya se habían obtenido los identificadores de las entidades de Wikidata, se pasó a hacer la consulta relacionada con los datos geográficos exactos de estas afiliaciones. Para este proceso se utilizó Wikidata y su servicio de consultas. La consulta se realizó en el lenguaje SPARQL y lo que se buscaba obtener eran los nombres de la ciudad y país donde está ubicada la afiliación, además de las coordenadas exactas de la afilición, la ciudad y el país respectivos. Esto se hizo a través de la siguiente consulta:

```
SELECT ?label ?wikidata (SAMPLE(?cityLabel) as ?sCity)
(SAMPLE(?headqLabel) as ?sHeadqCity) (SAMPLE(?countryLabel) as ?sCountryLabel)
(SAMPLE(?coord) as ?sCoord) (SAMPLE(?countryCoord) as ?sCountryCoord)
(SAMPLE(?cityCoord) as ?sCityCoord) (SAMPLE(?headqCoord) as ?sHeadqCoord)
WHERE {
    ?wikidata rdfs:label ?label .
    OPTIONAL {
        ?wikidata wdt:P625 ?coord .
    }
    OPTIONAL {
        ?wikidata wdt:P131 ?city .
        ?city rdfs:label ?cityLabel .
        ?city wdt:P625 ?cityCoord .
        FILTER (lang(?cityLabel) = "en")
    }
    OPTIONAL {
        ?wikidata wdt:P17 ?country .
        ?country rdfs:label ?countryLabel .
        ?country wdt:P625 ?countryCoord .
        FILTER (lang(?countryLabel) = "en")
    }
    OPTIONAL{
        ?wikidata wdt:P159 ?headq .
        ?headq rdfs:label ?headqLabel .
        ?headq wdt:P625 ?headqCoord .
         FILTER (lang(?headqLabel) = "en")
    }
    VALUES ?wikidata {...}
    FILTER (lang(?label) = "en")
}
GROUP BY ?wikidata ?label ?article
```

Código 3.7: Consulta en SPARQL para obtener datos geográficos de las afiliaciones.

Para entender esto, en la Tabla 3.2 se indican a qué corresponde la codificación de cada uno de los predicados presentes en la consulta. Es importante notar que en la línea 28 del ejemplo, donde hay tres puntos suspensivos, es donde se ubican todos los identificadores de las afiliaciones antes encontradas en el formato `wd:XXXXX` (para el ejemplo en Código 3.6 sería `wd:Q232141`). A su vez, es importante destacar que todas las consultas relacionadas con la ubicación geográfica van en una `OPTIONAL`. Esto es para garantizar que, aunque la entidad no tenga cada uno de esos atributos consultados, igual se obtenga en el resultado final de la consulta.

wdt	**Información del predicado**
P625	Coordenadas.
P131	Situado en la entidad territorial administrativa.
P159	Ubicación de la sede.
P17	País.

Tabla 3.2: Información sobre predicados en consulta SPARQL.

Una vez se realizó la consulta de SPARQL, se logró tener la información geográfica de las afiliaciones. Es importante destacar que este esquema es bastante irregular: algunas afiliaciones podrían tener una ciudad y un país asociado, mientras que otras podrían sólo tener un país. Además, para algunas afiliaciones Wikidata no provee las coordenadas exactas. Sin embargo, con estos resultados ya se tenían datos relevantes para comenzar a utilizar el sistema de base de datos, que es lo que se detallará en el siguiente capítulo.

Capítulo 4

Indexación y consulta de datos usando Elasticsearch

En el presente capítulo se detallará la construcción de la base de datos en sí, usando los resultados obtenidos que fueron mostrados en el Capítulo 3. Para explicar este proceso primero se hablará muy a grandes rasgos de cómo funciona Elasticsearch. Luego, se mencionará la configuración previa a la indexación y los campos a considerar de los datos de DBLP. Posterior a esto, se mostrará cómo se realizó el proceso de indexación más algunos ejemplos concretos de los datos que se almacenaron. Finalmente, se detallarán las consultas necesarias para obtener los resultados que se esperaba el sistema entregara.

4.1. Elasticsearch

Como se mencionó en el Capítulo 1, Elasticsearch es un motor de búsqueda que permite indexar documentos en formato JSON. Una unidad de indexación se llama índice (en inglés *index*). Estos documentos pueden tener ciertos campos de información, los cuales pueden ser diferentes entre uno y otro documento, a diferencia de un modelo relacional, donde todos los elementos de una tabla poseen los mismo campos. En este sentido, se entiende que Elasticsearch es libre de esquema. Es importante notar que un índice puede tener diferentes tipos de documentos, lo cual facilita el proceso de consulta al momento de filtrar. A su vez, un índice está organizado en distintos *shards* que permite la distribución y paralelización de operaciones sobre los datos. Además, el índice tiene un factor de réplica que permite recuperación de datos ante alguna eventualidad. A su vez, el motor de búsqueda está organizado en distintos nodos que finalmente conforman un *cluster*. Finalmente, es importante destacar que esta herramienta permite hacer todos sus procesos mediante una API REST, ya sea la indexación de datos, la consulta de ellos, la eliminación y edición.

Para el caso del trabajo se usó de un único índice, que estaba organizado en cinco *shards*, que era el valor por defecto. A su vez, se trabajó sólo con un único nodo, pues la cantidad de datos que había que indexar no era tan alta (el uso de más nodos está pensado y recomendado para un volumen mucho mayor de datos). Adicionalmente, se consideró usar dos tipos de documentos, esto debido a la naturaleza de los datos utilizados, que correspondían a **artículos**

de revista científica y **artículos de conferencia o workshop**.

Las consultas para el caso de Elasticsearch también se hacen usando el formato JSON. Lo interesante de este motor de búsqueda es que permite hacer muchos tipos de consultas, de distinta complejidad. Ejemplos de distintos tipos de consulta se pueden apreciar en Código 4.1 y 4.2. En el caso del ejemplo en Código 4.2 se explicita el número de documentos que se quieren obtener mediante la llave `size` (2 en ese caso).

```
{ "query":{
    "match_all": {}
  }
}
```

Código 4.1: Ejemplo de consulta simple de Elasticsearch. La consulta permite obtener todos los documentos indexados.

```
{
  "size":2
  "query": {
    "bool": {
      "must": [
        {"match":{
          "author": {"query": "alan turing"}
        }}
      ],
      "filter": [
        {"range":{"year":{"gte":2000}}}
      ]
    }
  }
}
```

Código 4.2: Ejemplo de consulta compleja de Elasticsearch. La consulta permite obtener todos los documentos filtrando sólo los que tengan en en el campo "year" un valor mayor a 2000 y tengan en el campo "author" "alan turing".

Es importante destacar que una de las características más atractivas de Elasticasearch es que permite hacer análisis de texto y poder hacer consultas de manera más eficiente o con mejores resultados desde el punto de vista del usuario. Dentro de estas características se encuentran el borrado de *stop words* (palabras de uso muy común), el uso de sinónimos a la hora de hacer consulta, el uso de *stemmer* (reducción de una palabra a su raíz), el uso de normalización de palabras modificable por usuario (por ejemplo, dejarlas en minúscula), entre otras.

Además un índice puede ser previamente configurado (mediante el uso de un archivo también en formato JSON) para que los procesos de indexación y consulta sean hechos de manera eficiente. Entre las configuraciones que se pueden hacer están la cantidad de *shards* a utilizar y el número de réplicas. También la definición de los analizadores mencionados

anteriormente puede hacerse mediante esta configuración previa, así como el *mapping* del esquema, es decir, la definición de los tipos de cada campo y qué analizador se debe usar en cada uno de ellos. A continuación se detallará cuáles fueron estas configuraciones para el caso del trabajo.

4.2. Configuración, *mapping* y campos de información

Antes de detallar la forma que fueron indexados los datos, es importante indicar cuáles fueron las configuraciones del índice, el *mapping* y los campos que se consideraron en cada artículo científico. Los campos a utilizar varían según si el artículo es de revista o de conferencia, sin embargo, ambos tienen algunos en común. Estos campos en común son los siguientes: título del artículo, autor(es), año de publicación, afiliación, coordenadas de la afiliación, ciudad, coordenadas de la ciudad, país, coordenadas del país y DOI. Para el caso de los artículos de revista se consideró el campo adicional para almacenar la revista en que se publicó. Es importante mencionar que las coordenadas se utilizaron en formato latitud y longitud, todo por separado, a diferencia del tipo de valor entregado por Wikidata, mostrado en Capítulo 3. Por otro lado, para los artículos de conferencia o workshop, adicionalmente se tienen los campos relacionados con el código de la conferencia o workshop, y el nombre del `proceedings` (colección de conferencia o workshop) asociado.

Como se mencionó anteriormente, se utilizaron las configuraciones por defecto para el índice, en cuanto a *shards*. Otras configuraciones importantes fueron las relacionadas a los distintos analizadores de texto. Entre ellos se consideró un filtro de palabras de tipo *stop words* y también el uso de *stemmer*, ambos procesos para el idioma inglés.

Para el caso de los *mappings*, se tuvo especial cuidado en los campos previamente definidos donde se haría la búsqueda de texto relacionada a los tópicos en ciencias de la computación. Estos campos eran: título del artículo, título de la revista científica o título de la conferencia o workshop, según corresponda. En este sentido, la definición de estos campos se basó en los analizadores antes establecidos en el mismo archivo de configuración. Una vez estaban definidos los *mappings* en el archivo de configuración, se procedió a crear el índice, el cual llevaría el nombre "papers", hecho mediante un comando de consola.

4.3. Indexación de datos

Con el índice creado ya sólo quedaba indexar los datos relacionados. Elasticsearch provee más de una opción para realizar este proceso, pero la más simple consiste en crear archivos en formato JSON e indexarlos mediante un comando de consola. Un ejemplo de un par de documentos presentes en un archivo JSON indexado se puede apreciar en Código 4.3 y 4.4.

```
{ "index" : {"_index" : "papers", "_type":"article", "_id":"447"} }
{
    "title": "Die Konzeption des EXCEPT-Systems: Ein \u00dcberblick",
    "authors": [
        "Martin H\u00fcbner",
        "Kai von Luck",
        "Ulrike Weiland"
    ],
    "journal": "IWBS Report",
    "year": 1990,
    "affiliationName": "University of Hamburg",
    "city": "Hamburg",
    "country": "Germany",
    "affLat": 53.56694444,
    "affLong": 9.98388889,
    "cityLat": 53.55,
    "cityLong": 10.0,
    "countryLat": 51.0,
    "countryLong": 10.0
}
```

Código 4.3: Ejemplo de artículo de revista añadido al índice `papers`.

```
{ "index" : {"_index" : "papers", "_type":"inproceedings", "_id":"879"} }
{
    "title": "Canonical Representations in Lisp
    and Applications to Computer Algebra systems.",
    "authors": ["Richard J. Fateman"],
    "booktitle": "ISSAC",
    "year": 1991,
    "doi": "https://doi.org/10.1145/120694.120750",
    "proceedingsName": "Proceedings of the 1991 International Symposium on Symbolic
    and Algebraic Computation, ISSAC '91, Bonn, Germany, July 15-17, 1991",
    "affiliationName": "University of California, Berkeley",
    "city": "Berkeley",
    "country": "United States of America",
    "affLat": 37.87,
    "affLong": -122.259,
    "cityLat": 37.870277777,
    "cityLong": -122.268055555,
    "countryLat": 39.828175,
    "countryLong": -98.5795
}
```

Código 4.4: Ejemplo de artículo de conferencia añadido al índice `papers`.

4.4. Consultas

Una vez los datos ya estaban disponibles en el índice, se comenzó a pensar en las consultas importantes que el sistema tendría que hacer. Como se ha mencionado a lo largo del informe, se esperaba que el sistema permitiera consultar sobre un tema de ciencias de la computación y entregara los datos de los artículos organizados según el lugar de la afiliación al cual están asociados. Con esto como objetivo, hace sentido utilizar el tipo de consulta de agregación que permite Elasticsearch. Para el caso de la futura visualización principal en un mapa se utiliza la consulta en Código 4.5, que busca el tópico que debe estar presente en el título (campo `title`), revista (`journal`) o colección de conferencia o workshop (`inproceedings`). Luego, en la parte inferior de la misma consulta, se hace la agregación (con la llave `aggs`) según la afiliación, ciudad, país y ambas componentes de sus coordenadas. Es importante notar que la consulta posee en su inicio el valor 0 para la llave `size`, esto con el motivo de que no se busca entregar ningún documento del índice que cumpla con la consulta, sino sólo las agregaciones definidas.

```
GET /papers/_search/
{
  "size":0,
  "query": {
     "bool": {
       "should": [
         {"match":{ "title": {"query":"semantic web", "operator": "and"}}},
         {"match":{ "proceedingsName": {"query":"semantic web", "operator":
         "and"}}},
         {"match":{ "journal": {"query":"semantic web", "operator": "and"}}}
       ],
       "minimum_should_match": 1
     }
  },
  "aggs": {
     "my_buckets": {
       "composite": {
           "size":1000,
           "sources": [
           {
               "Affiliation": {
               "terms": {
                   "field": "affiliationName.raw"
               }
               }
           },
           {
               "Lat": {
                 "terms": {
                     "field": "affLat"
                 }
               }
```

```
            },
            {
                "Long": {
                  "terms": {
                      "field": "affLong"
                  }
                }
            },
            {
                "Country": {
                  "terms": {
                      "field": "country.raw"
                  }
                }
            },
            {
                "City": {
                  "terms": {
                      "field": "city.raw"
                  }
                }
            }
        ]
      }
    }
  }
}
```

Código 4.5: Consulta para obtener datos desagregados según afiliación, consultando por el tema "semantic web".

La segunda consulta importante es para obtener todos los artículos asociados a una afiliación. En ese caso la consulta es similar, salvo que ya no se tiene la restricción del tamaño de la consulta y se añade como otro filtro el nombre de la afiliación que se está consultando. Un ejemplo de esto se puede apreciar en el Código 4.6. Se añade al final una agregación según el autor para de esta manera poder hallar cuál es el que tiene más artículos asociados en aquella afiliación. A su vez, a diferencia de la consulta anterior, se señalan al final cuáles son los campos de los documentos que se retornarán. Por otro lado, en comparación con la consulta mostrada anteriormente, aquí sí se requiere obtener explícitamente los documentos indexados, por lo que se señala el valor de la llave `"size"` como 1.000. Es importante destacar que Elasticsearch organiza sus documentos mediante paginación, por lo que si la cantidad de resultados de la consulta es mayor a 1.000, Elasticsearch permite acceder a los siguientes mil de manera eficiente.

```
GET /papers/_search/
{
  "size":1000,
  "query": {
     "bool": {
         "should": [
            {"match":{ "title": {"query":"semantic web", "operator": "and"}}},
            {"match":{ "proceedingsName": {"query":"semantic web", "operator":
            "and"}}},
            {"match":{ "journal": {"query":"semantic web", "operator": "and"}}}
         ],
         "minimum_should_match": 1,
         "filter": [
            { "term": { "affiliationName.raw": "University of Chile" }}
         ]
     }
  },
  "aggs":{
    "Authors":{
      "terms":{
         "field":"authors.raw"
      }
    }
  },
  "_source":{
    "includes": ["title","authors", "type", "year", "journal",
    "proceedingsName", "city", "country", "affiliationName", "doi"]
  }
}
```

Código 4.6: Consulta para obtener datos de los artículos en la afiliación University of Chile consultando por el tema semantic web.

Las consultas mostradas aquí son de vital importancia para el sistema en sí, pues son las que finalmente permitirán mostrar de manera adecuada los resultados esperados. El uso y modificación de estas consultas se detallarán en el Capítulo 5.

Capítulo 5

Frontend y experiencia de usuario

En el presente capítulo se detallará todo lo relacionado al *frontend* y la experiencia de usuario del sistema. Se mostrará primero una visión general de cómo se usó el *framework* Django y luego en distintas secciones se mostrarán las diferentes visualizaciones desarrolladas, señalando según sea el caso, las librerías relevantes que fueron utilizadas o las consultas que se hicieron al modelo para obtener los datos necesarios.

5.1. Django y visión general

Django es un *framework* del lenguaje de programación Python, que permite desarrollar aplicaciones web de manera simple, haciendo una separación de las distintas lógicas. Por un lado se tiene el modelo de datos (que en este caso corresponde a lo implementado en Elasticsearch), luego se tiene la lógica de las vistas, que se encarga de hacer el procesamiento y consulta de los datos y se comunica con las plantillas (desarrolladas en HTML), que muestran finalmente lo que un usuario visualiza. En este caso, las plantillas fueron desarrolladas mediante la librería Bootstrap, usando como base una preexistente llamada Start Bootstrap - Landing Page [13]. Para trabajar con Elasticsearch desde Django se utilizó una librería dedicada a hacer consultas sobre este motor de búsqueda. Estas consultas se pueden hacer usando el mismo formato JSON descrito en el Capítulo 4.

Es importante señalar que las distintas visualizaciones e interacciones con el sistema fueron pensadas poniendo énfasis en la simplicidad y que lograran ser muy intuitivas para el usuario. Se priorizó también usar una gama de colores sobrios y que fueran coherentes a través de todo el sistema.

5.2. Barra de navegación

La navegación del sistema se puede hacer a través de la barra superior presente en todas las interfaces. A través de ella se puede acceder a la página de inicio, realizar una búsqueda avanzada, ver toda la investigación del mundo relacionada con ciencias de la computación y acceder a un sitio donde se da información general relacionada con el sistema (*About*). A su vez, a un lado hay una caja de búsqueda que permite realizar una consulta con respecto a

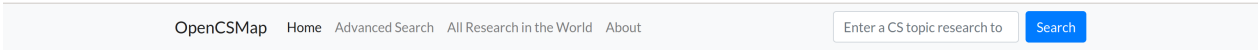

Figura 5.1: Barra de navegación del sistema.

un tema de ciencias de la computación.

5.3. Página principal y búsqueda simple

En la página principal (ver Figura 5.2) hay un breve mensaje que sintetiza la funcionalidad del sistema y una caja de búsqueda. En ella un usuario puede ingresar un tema de investigación para consultar sobre todos los artículos en ciencias de la computación relacionados a él. Al hacer esto, en el *backend*, el sistema realiza la consulta respectiva en Elasticsearch, la cual se ejemplifica en Código 4.5. Para el caso genérico, el tópico buscado por el usuario se ingresa donde, en el ejemplo, se halla el valor "semantic web". Al hacer eso se redirige a una visualización de un mapa, que organiza la información según las distintas afiliaciones encontradas.

Figura 5.2: Página principal.

5.4. Visualización en mapa de la investigación

En esta sección del sistema se muestran los resultados de la consulta anteriormente realizada (ver Figura 5.3). Inicialmente se despliega, en la parte superior, un resumen de la consulta, indicando el número de resultados totales, ya sea de artículos encontrados como del número de afiliaciones que publicaron sobre aquel tópico de investigación. Los resultados de las agregaciones se muestran a través de la librería Leaflet, desplegándose en un mapa de todo el mundo. Se usan los marcadores que provee esta librería, los cuales se ubican según las coordenadas (latitud y longitud) de la afiliación dada en los resultados de la consulta.

A su vez, se usa la opción de los conjuntos o *clusters*, que permite una visualización más limpia pues posibilita la agrupación de muchos marcadores cercanos en un sector geográfico. Al hacer un acercamiento estos *clusters* se van desagregando para dar la ubicación exacta de los marcadores. Además, un marcador tiene la opción de *pop-up*, donde se ven los nombres de las afiliaciones, la ciudad y país, y la cantidad de artículos que se hallaron para aquella afiliación. Al hacer click se redirige a una visualización que muestra todos los artículos sobre el tema, asociados a la afiliación.

Figura 5.3: Ejemplo de mapa generado con Leaflet para la consulta "semantic web". Se muestran *clusters*, marcadores y *pop-up*.

5.5. Artículos asociados a una afiliación

En esta visualización (ver Figura 5.4) se muestran todos los artículos sobre el tema consultado, asociados a una afiliación. Esto se logra mediante la consulta en Código 4.6. En el encabezado se muestra el tema consultado, la afiliación, el autor con más publicaciones en ella y la cantidad de artículos encontrados. Seguido a eso se muestra uno a uno todos los artículos que como resultado entrega la consulta. Se señala, para cada uno, el título de la publicación, los autores, la afiliación, en qué colección de conferencia o revista se publicó, el año y el DOI (el cual, de estar presente, redirige al vínculo del artículo). Como se indicó previamente, el esquema de la base de datos es libre, por lo que existe la posibilidad de que alguno de estos campos no esté disponible para algún artículo.

Figura 5.4: Ejemplo de un artículo asociado a una afiliación.

5.6. Búsqueda avanzada

Desde la barra de navegación se puede acceder a una búsqueda avanzada. A través de ella se pueden añadir filtros o búsquedas adicionales, presentes en un formulario (ver Figura 5.5), a la consulta sobre un tópico. Las opciones disponibles para una búsqueda avanzada son: indicar el autor de un artículo, rango de años de la publicación, organizar en el mapa los resultados según afiliación, ciudad o país y filtrar por tipo de publicación (de revista o conferencia). Desde el punto de vista de la consulta, ésta es muy parecida a la presentada en el Código 4.5, sin embargo se deben añadir filtros (usando la llave `filter`) para el caso de los tipos y los rangos de años, y, a su vez, se debe añadir una sección de consulta tipo `match` para el campo de autor. Para el caso de una agregación por ciudad o país, la sección de llave `aggs` de la consulta en Código 4.5 debe ser modificada. Si es por ciudad, se desestima el valor `Affiliation` y `affLat` y `affLong` deben ser reemplazados por `cityLat` y `cityLong`. Si es por país se hace el proceso análogo. Al realizar una búsqueda se obtiene una visualización similar a la del mapa anteriormente mostrado, salvo que el resumen del encabezado puede ser diferente según los filtros añadidos y, además, las agregaciones en el mapa pueden darse según ciudad o país (ver Figura 5.6).

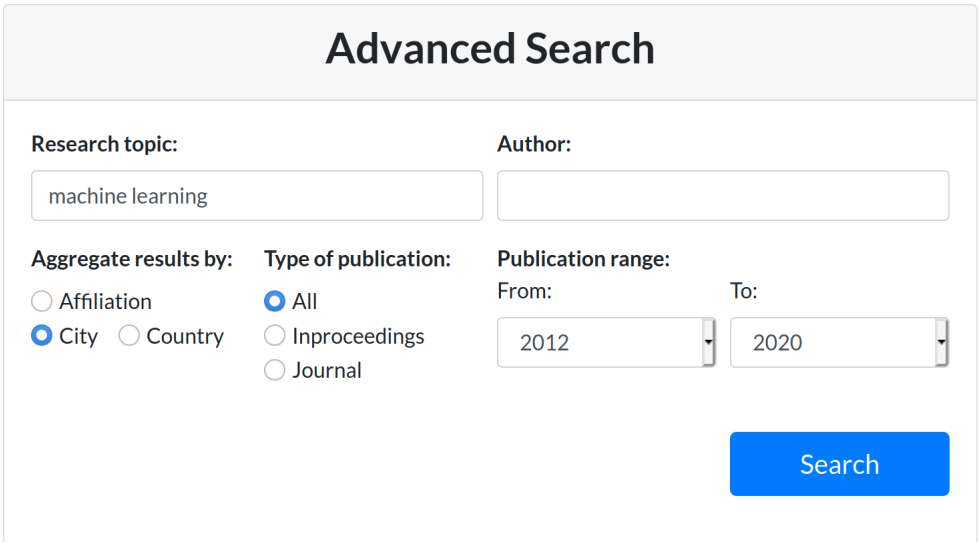

Figura 5.5: Formulario de búsqueda avanzada.

5.7. Toda la investigación en el mundo

Finalmente, también existe la posibilidad de ver el mismo mapa antes mostrado pero presentando todas las afiliaciones y los artículos geolocalizados en la base de datos. Se muestra el resumen (número total de artículos encontrados y de afiliaciones) en el encabezado. Esto tiene la misma funcionalidad que los mapas antes mencionados. Para obtener estos resultados basta con modificar la sección de llave `query` en la consulta en Código 4.5 por `"match_all":{}` (es decir, todos los documentos del índice).

El sistema desarrollado puede visitarse en `https://opencsmap.dcc.uchile.cl`. En el Capítulo 6 se mostrará una evaluación del sistema en general. Ésta se hará, por un lado, midiendo tiempos de respuesta para las distintas opciones de búsqueda y, por otro, a través de un cuestionario sobre la usabilidad del sistema.

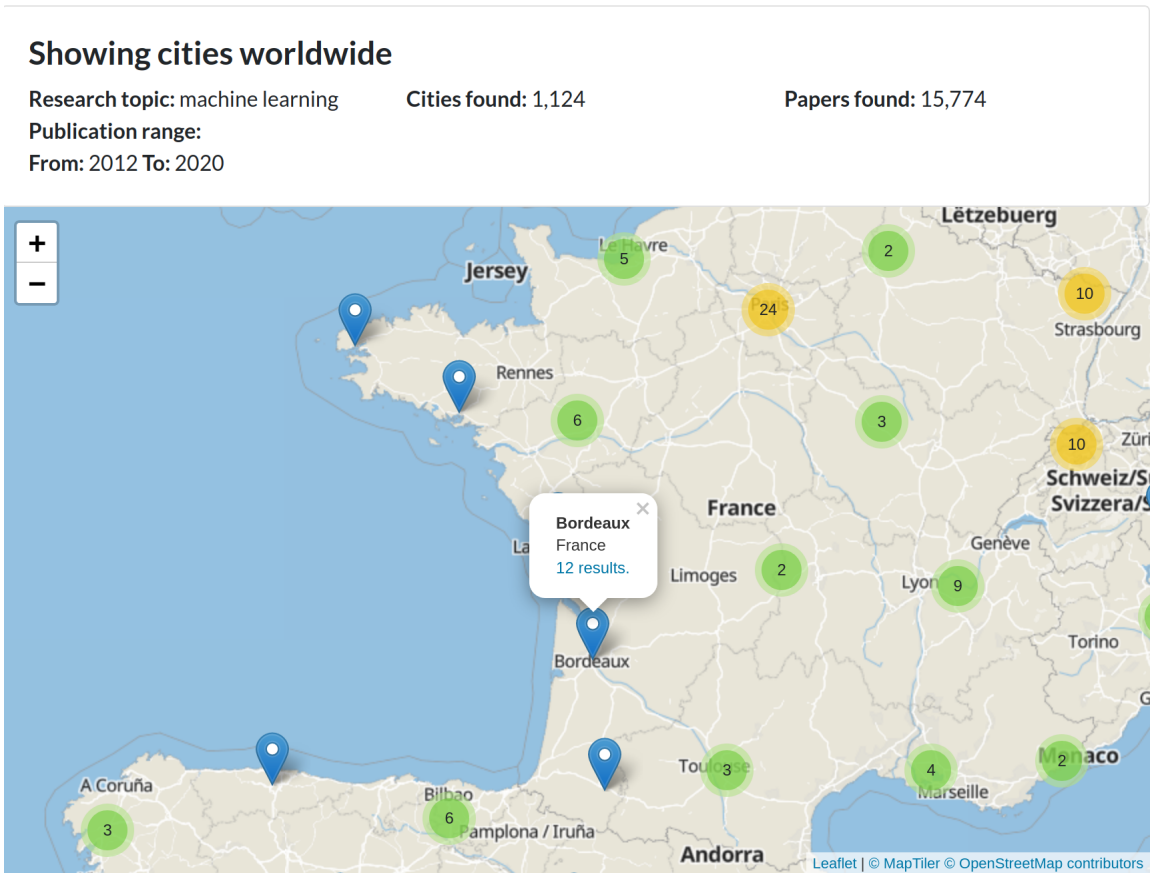

Figura 5.6: Resultados agregados por ciudad.

Capítulo 6

Evaluación

En el presente capítulo se hará una revisión de los distintos métodos usados para la evaluación del sistema. En primer lugar, se mostrará una evaluación del algoritmo usado para asignar afiliaciones a los artículos. Luego, se darán a conocer las pruebas realizadas para escoger la herramienta de *entity linking* utilizada para identificar las entidades relacionadas a las afiliaciones. Posteriormente, se detallará en distintas pruebas el uso y respuesta del sistema, principalmente desde una visión de tiempos de respuesta en distintos escenarios. Finalmente, se mostrarán los resultados obtenidos en un cuestionario aplicado a usuarios cuyo objetivo fue recopilar su percepción del sistema.

6.1. Algoritmo para asignar afiliaciones

Se realizó la evaluación del algoritmo para asignar afiliaciones, presentado en la Sección 3.2, con un un experimento para ver cuál fue la precisión de esta asignación. Este experimento consistió en seleccionar, de manera pseudoaleatoria, 100 artículos a los cuales se les intentó asignar una afiliación mediante el algoritmo. Luego, de manera manual, se buscó en la web uno a uno los artículos y se chequeó que la afiliación asignada por el algoritmo estuviera correcta. Como se mencionó en la Sección 3.2, el algoritmo sólo asigna una afiliación a un artículo, por lo que, en el experimento, se entiende como correcta una afiliación si corresponde a al menos una de todas las opciones del artículo. Para tener una mejor interpretación de la evaluación, se calcularon las medidas de precisión y *recall* (ver Tabla 6.1).

Precisión	***Recall***
0,72	0,38

Tabla 6.1: Resultados de evaluación de algoritmo para asignar afiliaciones.

Con estos valores, es importante notar que la precisión es bastante alta, es decir, un gran número de los artículos que se les intentó asignar una afiliación, se logró satisfactoriamente. Sin embargo, el *recall* tiene un valor mucho menor, esto es debido a que, como se mencionó en la Sección 3.2, hubo un gran número de artículos que el algoritmo no pudo asignarle una afiliación (cerca del 50 %).

6.2. Herramientas de *entity linking*

Otro aspecto del sistema que fue necesario evaluar fueron las herramientas de *entity linking*. Las herramientas que fueron consideradas para poder identificar las entidades de las afiliaciones fueron: DBpedia Spotlight, TAGME y OpenTapioca. Para evaluar cada una de ellas, se tomaron 100 afiliaciones que debían ser identificadas y se vio el desempeño de cada una de las herramientas antes mencionadas. En este caso, a diferencia de la evaluación del algoritmo de asignación de afiliaciones, se calculó sólo la precisión pues para todos los casos todas las herramientas al menos identificaron una entidad. Los resultados de la precisión se pueden observar en la Tabla 6.2.

DBpedia Spotlight	**TAGME**	**OpenTapioca**
0,48	0,81	0,84

Tabla 6.2: Resultados de precisión de herramientas de *entity linking*.

Como se puede observar en los resultados, OpenTapioca fue la herramienta que obtuvo mayor precisión. Sin embargo, ésta no tuvo mucha diferencia en comparación con TAGME. Lo importante, y tal como se explicó en la Sección 3.3, es que OpenTapioca fue escogida no sólo por su precisión. En primer lugar, como se utilizaría Wikidata para obtener los datos geográficos de las afiliaciones, resultaba muy conveniente que la respuesta de OpenTapioca fuera el identificador de esta base de conocimiento, en comparación con TAGME, cuya respuesta es el nombre de la entidad de Wikipedia, lo cual generó cierta complejidad al momento de consultar a Wikidata, sobre todo con el uso de caracteres especiales. A su vez, el uso de OpenTapioca al hacer el vínculo con Wikidata, era más promisorio porque la base de conocimientos cuenta con muchas más entidades que Wikipedia, que es la fuente que usa TAGME para hacer los vínculos.

Por otro lado, es importante destacar el bajo valor resultante para DBpedia Spotlight, en comparación con TAGME y OpenTapioca. Al chequear en detalle las respuestas dadas por esta herramienta, se pudo observar que ésta fallaba mucho para afiliaciones correspondientes a universidades. Por ejemplo, de la afiliación "University of California" reconocía las entidades "University" y "California", pero no la que conforman ambas palabras en su conjunto. Al ser muy común el uso de la palabra "University" en el conjunto de afiliaciones, era muy notorio el fallo para esta herramienta.

6.3. Pruebas de rendimiento del sistema

Para medir el rendimiento del sistema, se diseñaron algunas pruebas. Estas pruebas se separaron en dos categorías. En primer lugar se quiso evaluar el desempeño que por sí sólo daba Elasticsearch, el motor de búsqueda utilizado en el sistema. Por otro lado, era también importante evaluar, en escenarios lo más cercanos posibles a la realidad, el sistema en general. Para las dos categorías de pruebas mencionadas, se observaron tres escenarios distintos: la búsqueda simple de un tema de investigación de ciencias de la computación, la búsqueda avanzada que provee el sistema y, finalmente, búsquedas simples ejecutadas en paralelo. Las búsquedas en paralelo se realizaron para poder simular el uso concurrente que se le pudiera dar al sistema. Por otro lado, para poder hacer las búsquedas se utilizaron las palabras clave

encontradas en el conjunto de datos de Open Academic Graph. Como se mencionó en la Sección 2.1, OAG provee palabras clave que usualmente se encuentran en los encabezados de los artículos científicos. Al hacer la intersección con DBLP, para así obtener sólo artículos de ciencias de la computación, se encontraron 1.552.529 palabras clave distintas. De ellas, para estas pruebas, sólo se utilizaron 5.000, que correspondían a las encontradas con mayor frecuencia en los artículos de ciencias de la computación. A continuación se detallará en qué consistieron las pruebas realizadas.

6.3.1. Pruebas sobre Elasticsearch

Las pruebas sobre Elasticsearch fueron realizadas independientemente del sistema, es decir, se hicieron las consultas de manera directa al motor de búsqueda. Para el caso de una búsqueda simple, se realizaron 1.000 consultas de forma secuencial, sobre alguna palabra clave seleccionada de manera aleatoria, obteniendo el tiempo de ejecución de cada una de ellas y la cantidad de afiliaciones que cada búsqueda entregaba como resultado. En la Tabla 6.3 se pueden observar el promedio del tiempo obtenido en milisegundos (A), la desviación estándar del tiempo en milisegundos (B), el promedio de afiliaciones obtenidas por búsqueda (C) y su desviación estándar (D).

A	**B**	**C**	**D**
18,28	11,99	473,27	386,33

Tabla 6.3: Resultados de pruebas de búsqueda simple en Elasticsearch.

Para el caso de una búsqueda avanzada se realizaron también 1.000 consultas en base a una palabra clave obtenida de forma aleatoria. Este tipo de búsqueda necesita más valores (autor, tipo de publicación, de qué manera los resultados debieran ser agregados y rango de años), por lo que éstos también fueron escogidos de manera aleatoria. Para el caso del autor, para evitar una gran cantidad de búsquedas con resultados vacíos, se decidió sólo usar el nombre de pila de los autores que se pueden hallar en el conjunto de datos de DBLP. En este caso se registraron (ver Tabla 6.4) el tiempo promedio en milisegundos (A), su desviación estándar en milisegundos (B), el promedio de cantidad de afiliaciones encontradas (C), su desviación estándar (D) y la cantidad de búsquedas que entregaron resultados vacíos (E).

A	**B**	**C**	**D**	**E**
8,44	4,94	57,31	149,99	452

Tabla 6.4: Resultados de pruebas de búsqueda avanzada en Elasticsearch.

Finalmente, para el caso de las búsquedas simples en paralelo, se realizaron tres experimentos diferentes: uno con 2 hilos, otro con 3 y finalmente uno con 4. Para cada uno de esos experimentos, cada hilo realizaba 1.000 consultas sobre un tópico escogido aleatoriamente. Cada experimento se ejecutó 5 veces y se escogió la mediana entre los tiempos totales de ejecución. En la Figura 6.1 se pueden observar los resultados obtenidos, donde se muestra el tiempo total de ejecución de la prueba, según la cantidad de hilos. Es importante notar que las pruebas con 2, 3 y 4 hilos realizaron, respectivamente, 2000, 3000 y 4000 búsquedas en total.

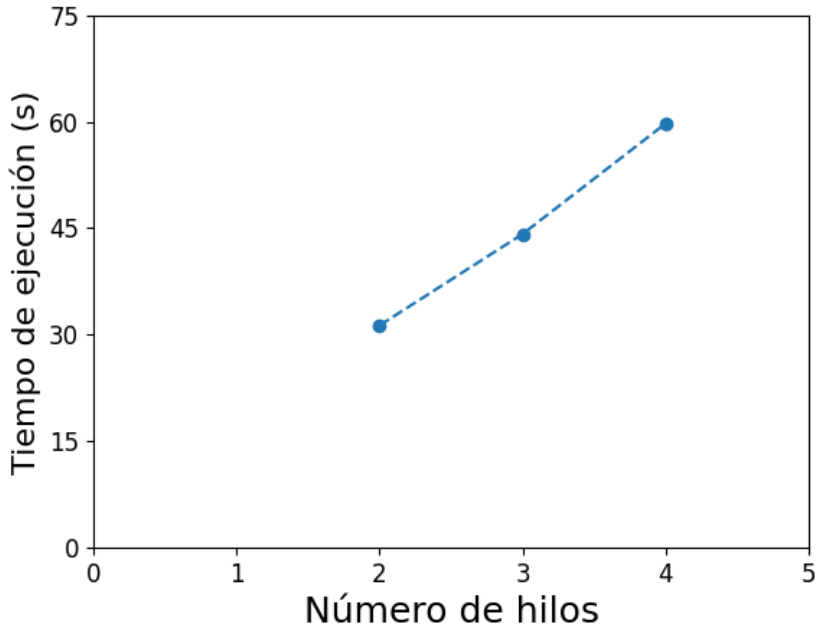

Figura 6.1: Tiempos de ejecución para pruebas en paralelo en Elasticsearch.

Para estos experimentos es importante destacar varios aspectos. En primer lugar, para una búsqueda simple, que debiera ser la más común dentro del sistema, los tiempos son bastante bajos y, si bien se presenta una dispersión no menor, al ser tiempos del orden de milisegundos esto no debiese significar una variación muy significativa. Por otro lado, para el caso del experimento de búsqueda avanzada, estos tiempos fueron menores en comparación con la búsqueda simple. Esto, teniendo en cuenta los resultados de ambos experimentos, puede deberse a que, por un lado, el número promedio de resultados de afiliaciones es mucho mayor en el experimento de una búsqueda simple, y por otro, que tal como se muestra en la Tabla 6.4 muchas de las búsquedas avanzadas no obtuvieron resultado alguno. De esto se podría deducir, entonces, que los tiempos de respuesta se relacionan más con la cantidad de resultados que el motor de búsqueda entrega y no tanto con el nivel de complejidad de la consulta.

Para el caso de los experimentos en paralelo, como se evidencia en la Figura 6.1, no se logra observar un paralelismo real. La curva tiene un comportamiento lineal que sugiere que a medida que hay más consultas realizadas simplemente aumenta el tiempo de ejecución. Sin embargo, la cantidad de consultas fue alta en comparación con un uso en condiciones normales de usuarios y, sin embargo, los tiempos de ejecución totales no fueron particularmente elevados, por lo que se cree que aun así hay resultados razonables de respuesta.

6.3.2. Pruebas sobre el sistema en general

Para evaluar el sistema en general se realizaron experimentos con las mismas condiciones de la Subsección 6.3.1. La única diferencia es que en el caso de estas pruebas se trató de simular el uso real de un usuario mediante peticiones HTTP. Los resultados son análogos a los mostrados en la subsección anterior. En la Tabla 6.5 se pueden apreciar los resultados

para 1.000 búsquedas simples, con el tiempo promedio de una búsqueda (A), su desviación estándar (B), la cantidad de afiliaciones halladas (C) y su desviación estándar (D).

A	B	C	D
147,18	93,11	464,049	377,29

Tabla 6.5: Resultados de pruebas de búsqueda simple en sistema general.

En la Tabla 6.6 se encuentran los resultados obtenidos para la búsqueda avanzada. Se registran, nuevamente, el promedio de tiempo de ejecución de una búsqueda (A), su desviación estándar asociada (B), número promedio de afiliaciones halladas (C) con su desviación estándar (D) y el número total de búsquedas que no entregaron resultados (E).

A	B	C	D	E
53,39	39,16	59,693	153,93	459

Tabla 6.6: Resultados de pruebas de búsqueda avanzada en sistema general.

En cuanto a las pruebas realizadas en paralelo, se pueden apreciar los resultados en la Figura 6.2. Para este caso, a diferencia del antes mostrado, cada hilo ejecutó 250 búsquedas, para así obtener tiempos razonables en las pruebas. También se ejecutaron 5 veces los experimentos y se escogió la mediana como el resultado a mostrar.

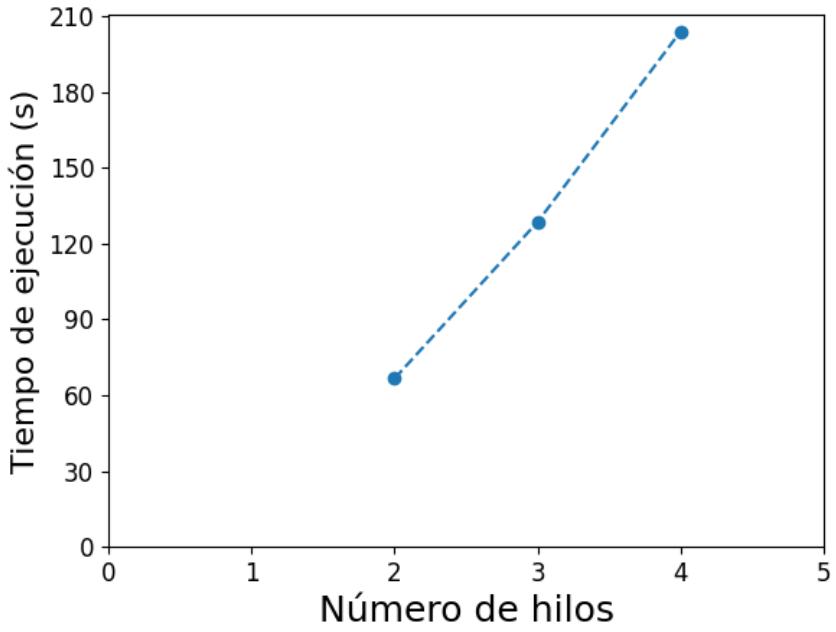

Figura 6.2: Tiempos de ejecución para pruebas en paralelo en sistema general.

Con respecto a estas pruebas, es importante destacar que los tiempos para los tres tipos de experimentos fueron mayores a las pruebas análogas ejecutadas sólo sobre Elasticsearch. Sin embargo, tanto para las pruebas de búsqueda simple y avanzada, el orden del promedio sigue siendo de los milisegundos, por lo que aun cuando la petición HTTP supone un trabajo mayor, los tiempos de respuesta siguen siendo bajos.

Por otro lado, para las pruebas realizadas en paralelo, se ve una cierta similitud en la curva de la Figura 6.2 para las pruebas del sistema general y la Figura 6.1 de las pruebas sólo sobre Elasticsearch: ambas tienden a una forma lineal. Sin embargo, llama la atención que aun cuando las pruebas para el sistema general implicaban menos consultas realizadas por hilo (250 en comparación con 1000 para pruebas sólo sobre Elasticsearch), los tiempos fueron mucho mayores. Esto puede deberse a que las peticiones HTTP requieren más tiempo en comparación al tiempo de consulta sobre Elasticsearch. Aún así, es importante señalar que nuevamente la cantidad de consultas era bastante y los tiempos no fueron tan altos, por lo que se cree que los resultados de esta prueba son aceptables. Aun así, es importante, en un futuro, poner énfasis en buscar soluciones para lograr un paralelismo real al momento de hacer consultas.

6.4. Percepción de usuarios

Para ver cuál fue la percepción de los usuarios, se creó un cuestionario que fue compartido, junto con el enlace del sistema, a docentes pertenecientes al Departamento de Ciencias de la Computación de la Universidad de Chile. El cuestionario consistió en 6 afirmaciones con respuesta de escala Likert y 3 preguntas de respuesta libre. Respectivamente, fueron las siguientes:

1. Entiendo el propósito del sistema.
2. Me parece intuitivo el uso del sistema.
3. El sistema provee algo novedoso con respecto a otros sistemas disponibles en la web.
4. Utilizaría este sistema, una vez esté más desarrollado.
5. Creo que los datos que vi en el sistema son precisos y completos.
6. Creo que el tiempo de respuesta de las búsquedas es razonable.
7. Indique a continuación los elementos positivos que ve en este sistema.
8. Indique a continuación los elementos negativos que ve en este sistema.
9. Indique a continuación los elementos que cree podrían ser añadidos al sistema.

El cuestionario recibió 10 respuestas en total. En la Tabla 6.7 se puede apreciar la distribución de respuestas de las preguntas 1 a la 6, donde L1, L2, L3, L4 y L5 corresponden a las respuestas de la escala Likert "muy en desacuerdo", "en desacuerdo", "neutro", "de acuerdo" y "muy de acuerdo", respectivamente. En general, las respuestas fueron positivas en cuanto a las preguntas relacionadas con la usabilidad del sistema: se entendía el propósito (pregunta 1) y el uso parecía intuitivo (pregunta 2). A su vez, la pregunta 3 relacionada con la novedad del sistema, en comparación con otros disponibles en la web, también recibió respuestas favorables. Además, la pregunta 6 que apuntaba a los tiempos de respuesta del sistema tuvo en su mayoría buenos resultados. Sin embargo, las preguntas 4 y 5, que correspondían a si usarían el sistema a futuro y si los datos que se vieron eran precisos, respectivamente, recibieron respuestas más variadas. De aquello se puede interpretar que el uso que se le puede dar al sistema tal vez no es de tanto interés o no se le ve un potencial muy grande y, por otro lado, que los datos que muestra el sistema no son del todo relevantes o completos (situación ya mencionada a lo largo del informe).

En cuanto a las respuestas de las preguntas abiertas (7, 8 y 9), se obtuvieron comentarios

Preguntas/Likert	**L1**	**L2**	**L3**	**L4**	**L5**
P1	1	0	2	4	3
P2	1	0	1	7	1
P3	1	0	3	4	2
P4	2	1	2	3	2
P5	2	3	2	3	0
P6	1	0	0	2	7

Tabla 6.7: Resultados de cuestionario sobre percepción de usuarios.

muy variados. En cuanto a los aspectos positivos, la mayoría apuntaban a temas relacionados con la interfaz, mencionando que era fácil de usar, que era innovadora y que permitía rápidamente encontrar investigación en algún sector del mapa. Para los comentarios negativos, la mayoría estuvieron enfocados en las imprecisiones en cuanto a tópicos buscados como en la información geográfica. Otros comentarios eran sobre algunas visualizaciones que no eran muy intuitivas, el uso de la búsqueda avanzada, por ejemplo. En cuanto a las recomendaciones, se señalaron aspectos como hacer vínculos con los sitios asociados a autores de DBLP, para así poder observar otros artículos de algún autor de interés. También se recibió la recomendación de usar Semantic Scholar como alternativa de fuente de datos.

Algunas de estas evaluaciones servirán para poder ver el logro de los objetivos inicialmente planteados. Estos se revisarán en el Capítulo 6.4.

Conclusión

A lo largo del trabajo se habló sobre distintos aspectos que luego, en su conjunto, se convirtieron en un sistema que pudiera consultar sobre la geolocalización en la investigación en ciencias de la computación. Por un lado, se trabajó con datos de ciencias de la computación (principalmente de artículos científicos) que debieron ser geolocalizados, para lo cual se utilizaron herramientas como OpenTapioca y el servicio de consultas de Wikidata. En una segunda parte del trabajo, se diseñó el modelo de datos. Como motor de búsqueda se utilizó Elasticsearch, que permitió realizar consultas de texto en relación a tópicos de investigación de manera muy eficiente. Finalmente, se desarrollaron las interfaces y se pensó la experiencia de usuario. La visualización principal del sistema consiste en un mapa que señala afiliaciones donde se haya desarrollado algún escrito sobre un tema de ciencias de la computación. Al final de todo este proceso, se obtuvo como resultado un sistema usable, que tiene buenos tiempos de respuesta a consultas y que logra mostrar una visión de cómo se da la investigación de las ciencias de la computación en todo el mundo.

Una manera de evaluar los resultados finales del trabajo realizado, es analizar los objetivos planteados en un inicio. El objetivo general del trabajo era: "desarrollar un sistema que permita realizar búsquedas sobre investigación en ciencias de la computación, agrupando los resultados según el lugar de producción y visualizando éstos". Se cree que este objetivo se logró, pues se pudo desarrollar un sistema con las características mencionadas.

Por otro lado, los objetivos específicos que apoyaban el proceso de cumplir el objetivo general eran cuatro. El primero consistía en obtener datos relacionados con ciencias de la computación, el cual se logró pues se pudieron utilizar los provistos por DBLP y se exploraron otras alternativas como las de Open Academic Graph y Semantic Scholar. El segundo objetivo específico era desarrollar un método para organizar artículos según geolocalización, lo cual se cree se pudo cumplir medianamente. Si bien, como se mencionó en el Capítulo 3, se pudo desarrollar un método que lograra asignar afiliaciones a artículos (evaluado en la Sección 6.1) y se pudo geolocalizar estas afiliaciones utilizando OpenTapioca y Wikidata, hubo un gran número de estos artículos a los cuales no se les pudo asignar una localización. A su vez, este método tiene limitaciones (como que asignaba sólo una afiliación por artículo o que no considera la trayectoria académica de los autores) por lo que uno de los desafíos a futuro es poder solucionar estos problemas.

El tercer objetivo consistía en desarrollar un método que pudiera organizar publicaciones según su tema de investigación. Aun cuando se pudo tener una alternativa utilizando Elasticsearch para hacer búsqueda de texto sobre distintos tópicos, esto no solucionó el problema

descrito y, por ende, este objetivo no pudo ser satisfecho. Como se mencionó anteriormente, se intentó utilizar datos y herramientas ya existentes, sin embargo, éstas no lograron dar resultados para un porcentaje significativo de los datos utilizados (para el caso de Open Academic Graph) o su uso no era compatible con el tiempo en que se enmarcaba el trabajo (CSO Classifier). El cuarto y último objetivo planteaba desarrollar visualizaciones amigables para los usuarios. Se cree que este objetivo sí fue cumplido pues, por un lado, las respuestas al cuestionario de percepción de usuarios, detalladas en la Sección 6.4, eran mayoritariamente positivas en lo que respecta a este tema y, por otro lado, como se mostró en la Sección 6.3, los tiempos de respuesta del sistema eran buenos (del orden de los milisegundos) lo cual también se puede traducir en una buena experiencia de usuario.

Finalmente, es importante mencionar que se espera poder seguir trabajando en el sistema a futuro. Si bien lo realizado durante este trabajo de título fue un buen comienzo, aún quedan algunas fuentes de datos que explorar (como ArnetMiner) y herramientas que podrían ser de utilidad. Un aspecto que no pudo ser tratado de manera exhaustiva, fue lo relacionado con los temas de investigación. Se espera poder poner énfasis en esto en el futuro, ya sea utilizando a cabalidad la herramienta CSO Classifier o diseñando algún modelo de aprendizaje automático que pueda clasificar los artículos según su tópico. Por otro lado, un punto importante fue el problema del paralelismo al hacer consultas en el sistema. En este sentido, hubo muchas características de Elasticsearch que no fueron revisadas a lo largo del trabajo y se espera poder explotar todo el potencial pueda ofrecer en relación a esto. Todas estas son ideas que se espera puedan ir en la dirección de mejorar el sistema desarrollado y éste se pueda convertir en el futuro en una herramienta útil para los distintos investigadores del área.

Bibliografía

[1] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew E. Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In Srinivas Bangalore, Jennifer Chu-Carroll, and Yunyao Li, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 3 (Industry Papers)*, pages 84–91. Association for Computational Linguistics, 2018.

[2] Andrzej Bialecki, Robert Muir, and Grant Ingersoll. Apache lucene 4. In Andrew Trotman, Charles L. A. Clarke, Iadh Ounis, J. Shane Culpepper, Marc-Allen Cartright, and Shlomo Geva, editors, *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval, OSIR@SIGIR 2012, Portland, Oregon, USA, 16th August 2012*, pages 17–24. University of Otago, Dunedin, New Zealand, 2012.

[3] Alexandru Boicea, Florin Radulescu, and Laura Ioana Agapin. Mongodb vs oracle - database comparison. In *2012 Third International Conference on Emerging Intelligent Data and Web Technologies, Bucharest, Romania, September 19-21, 2012*, pages 330–335. IEEE Computer Society, 2012.

[4] Kristina Chodorow and Michael Dirolf. *MongoDB - The Definitive Guide: Powerful and Scalable Data Storage*. O'Reilly, 2010.

[5] Antonin Delpeuch. Opentapioca: Lightweight entity linking for wikidata. *CoRR*, abs/1904.09131, 2019.

[6] Catherine D'Ignazio, Rahul Bhargava, Ethan Zuckerman, and Luisa Beck. Cliff-clavin: Determining geographic focus for news. *NewsKDD: Data Science for News Publishing, at KDD*, 2014, 2014.

[7] Paolo Ferragina and Ugo Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1625–1628. ACM, 2010.

[8] Mordechai (Muki) Haklay and Patrick Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Comput.*, 7(4):12–18, 2008.

[9] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[10] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[11] Michael Ley. DBLP - some lessons learned. *Proc. VLDB Endow.*, 2(2):1493–1500, 2009.

[12] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In Chiara Ghidini, Axel-Cyrille Ngonga Ngomo, Stefanie N. Lindstaedt, and Tassilo Pellegrini, editors, *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM, 2011.

[13] David Miller, Pavel Nikitin, and Nil Weerasinghe. startbootstrap-landing-page. `https://https://github.com/startbootstrap/startbootstrap-landing-page`.

[14] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Trans. Assoc. Comput. Linguistics*, 2:231–244, 2014.

[15] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. pigeo: A python geotagging tool. In Sameer Pradhan and Marianna Apidianaki, editors, *Proceedings of ACL-2016 System Demonstrations, Berlin, Germany, August 7-12, 2016*, pages 127–132. Association for Computational Linguistics, 2016.

[16] Angelo Antonio Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. The CSO classifier: Ontology-driven detection of research topics in scholarly articles. In Antoine Doucet, Antoine Isaac, Koraljka Golub, Trond Aalberg, and Adam Jatowt, editors, *Digital Libraries for Open Knowledge - 23rd International Conference on Theory and Practice of Digital Libraries, TPDL 2019, Oslo, Norway, September 9-12, 2019, Proceedings*, volume 11799 of *Lecture Notes in Computer Science*, pages 296–311. Springer, 2019.

[17] Angelo Antonio Salatino, Thiviyan Thanapalasingam, Andrea Mannocci, Francesco Osborne, and Enrico Motta. The computer science ontology: A large-scale taxonomy of research areas. In Denny Vrandecic, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II*, volume 11137 of *Lecture Notes in Computer Science*, pages 187–205. Springer, 2018.

[18] Guillermo Peralta Scura. vue-choropleth. `https://github.com/voluntadpear/vue-choropleth`.

[19] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (MAS) and applications. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 243–246. ACM, 2015.

[20] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 990–998. ACM, 2008.

[21] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

[22] Gong-Qing Wu, Ying He, and Xuegang Hu. Entity linking: An issue to extract corresponding entity with knowledge base. *IEEE Access*, 6:6220–6231, 2018.

[23] Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. AIDA: an online tool for accurate disambiguation of named entities in text and tables. *Proc. VLDB Endow.*, 4(12):1450–1453, 2011.