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

MEJORANDO LOS FLUJOS DE DATOS DEL IMFD

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

MATÍAS RODRIGO RIVERA CONTRERAS

PROFESOR GUÍA:
AIDAN HOGAN

PROFESORA CO-GUÍA:
CAMILA DÍAZ FOXON

MIEMBROS DE LA COMISIÓN:
MARÍA CECILIA BASTARRICA PIÑEYRO
HUGO MORA RIQUELME

SANTIAGO DE CHILE
2025

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN
POR: MATÍAS RODRIGO RIVERA CONTRERAS
FECHA: 2025
PROF. GUIA: AIDAN HOGAN

MEJORANDO LOS FLUJOS DE DATOS DEL IMFD

El Instituto Milenio Fundamentos de los Datos (IMFD), centro de excelencia en ciencia de datos en Chile, enfrentaba desafíos significativos en la gestión de su propia información institucional. A pesar de contar con una plataforma basada en Wikibase para gestionar datos sobre publicaciones e investigadores, esta permanecía subutilizada, mientras la información crítica se encontraba fragmentada en múltiples formatos y ubicaciones, generando duplicación, inconsistencias y una considerable inversión de tiempo en tareas manuales de recolección.

Esta memoria presenta el diseño e implementación de un sistema automatizado que centraliza y optimiza los flujos de información bibliográfica del IMFD. La solución integra cuatro fuentes de datos académicos principales (OpenAlex, DBLP, Web of Science y Scopus) mediante un *pipeline* de extracción, transformación y carga orquestado en Apache Airflow. El sistema procesa semanalmente las publicaciones de los investigadores, enriquece los metadatos desde múltiples fuentes y actualiza automáticamente el grafo de conocimiento institucional basado en Wikibase.

La implementación incluye mecanismos de actualización incremental que procesan únicamente publicaciones modificadas, un manejo robusto de errores y estrategias para evitar duplicación de entidades. Se desarrolló además un plugin de WordPress que conecta el grafo de conocimiento con el sitio web institucional, permitiendo la visualización pública actualizada de las publicaciones científicas.

La evaluación del sistema demostró su efectividad en tres dimensiones clave: la integración exitosa con el sitio web institucional, la generación automatizada de reportes comparables con las memorias anuales históricas y una puntuación de 81/100 en pruebas de usabilidad SUS, confirmando que el sistema resulta intuitivo para los usuarios finales.

Este trabajo establece una infraestructura robusta para la gestión del conocimiento institucional, reduciendo significativamente el esfuerzo manual requerido y mejorando la accesibilidad de la información científica del IMFD, mientras sienta las bases para futuros desarrollos en la automatización de procesos académicos.

Para Violeta y Cecilia.

Agradecimientos

Agradezco con mucho cariño a mi mamá y abuelita, por inculcarme valores como la perseverancia y la dedicación, además de darlo todo para que nunca me faltara nada.

Gracias a la Rena, por siempre escucharme hablar de la memoria, mostrar asombro y motivarme a seguir avanzando aún cuando los desafíos me hacían dudar. Gracias por estar ahí.

También le doy las gracias a mis roomies, Pame y Brayan, por brindarme compañía en mi vida universitaria, siendo un cable a tierra después de días agotadores.

No puedo dejar fuera a los gatos. Cata, Romi, Maca y Darlene. Sin duda el tiempo fue más ameno y divertido gracias a ustedes. Nunca olvidaré los planes para ir al Dunkin, los paseos y el grato espacio que compartimos.

También quiero mencionar al Franz, partimos siendo compañeros en un ramo pero no nos quedamos solo con eso. Gracias por mostrarme tus intereses y por ser un gran amigo.

Finalmente quiero agradecer a Aidan, por darme la confianza que necesitaba durante el desarrollo de la memoria y hacerme ver que iba avanzando bien.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Índice de Códigos

Capítulo 1

Introducción

1.1. Contexto

En la era digital actual, los datos se han consolidado como uno de los recursos más valiosos para organizaciones de todo tipo. La capacidad de extraer valor significativo de estos datos no solo permite generar nuevos conocimientos, sino que también impacta de manera directa en la toma de decisiones estratégicas y la gestión operativa de instituciones tanto públicas como privadas. Sin embargo, el creciente volumen y variedad de los datos modernos presentan desafíos complejos que requieren soluciones tecnológicas innovadoras para su gestión.

En este contexto, el Instituto Milenio Fundamentos de los Datos (IMFD) emerge como un centro científico multidisciplinario de excelencia en Chile, dedicado a la investigación y desarrollo en ciencia de datos. El IMFD aborda de manera integral los desafíos relacionados con el ciclo completo de vida de los datos, desde su obtención y procesamiento hasta su análisis e impacto social. Esta iniciativa representa una colaboración estratégica entre la Pontificia Universidad Católica de Chile y la Universidad de Chile, con la participación activa de académicos de diversas universidades nacionales y cuenta con el respaldo financiero de la Iniciativa Científica Milenio.

Paradójicamente, una institución que se dedica profesionalmente a la extracción de valor desde los datos enfrenta desafíos significativos en la gestión de su propia información institucional. Actualmente, el IMFD cuenta con una plataforma basada en Wikibase[1] para gestionar información sobre publicaciones asociadas a sus investigadores, tesis y memorias. Sin embargo, esta herramienta no ha logrado integrarse efectivamente en los flujos de trabajo del Instituto, manteniéndose subutilizada y con información desactualizada.

[1] https://wikiba.se/

1.2. Problema y Relevancia

El problema central surge de la fragmentación y dispersión de datos relacionados con las actividades del Instituto, que se encuentran distribuidos en múltiples formatos y ubicaciones, como en hojas de cálculo de Google Sheets, documentos PDF, archivos Excel y bases de datos asociadas al sitio web institucional. Esta fragmentación genera consecuencias operativas significativas que incluyen duplicación de datos, información desactualizada e inconsistente, además que involucra procesos manuales propensos a errores humanos y una considerable inversión de tiempo en tareas de recolección y consolidación de datos. Los flujos de información provienen de diversas fuentes, incluyendo las investigaciones y proyectos desarrollados por sus académicos, así como las actividades de divulgación científica gestionadas por el equipo de comunicaciones, que reportan sobre eventos, proyección mediática y entrevistas.

Esta problemática es particularmente relevante considerando que el IMFD debe generar regularmente contenido actualizado para su sitio web, elaborar memorias anuales institucionales y proporcionar reportes de gestión a organismos externos. La falta de un sistema centralizado y automatizado no solo impacta la eficiencia operativa, sino que también limita la capacidad del Instituto para aprovechar la información disponible para conocer su estado actual, realizar monitoreo continuo de actividades y abrir oportunidades para análisis más sofisticados, como la identificación de posibles colaboraciones entre investigadores con intereses en común o visualizar el impacto de las publicaciones científicas.

Es necesario que una institución que trabaja en estrecha relación con datos, tenga una gestión eficiente de sus propios recursos de información, garantizando su accesibilidad y precisión.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar e implementar un sistema automatizado de gestión de datos que centralice, actualice y optimice los flujos de información del Instituto Milenio Fundamentos de los Datos, integrando efectivamente el grafo de conocimiento basado en Wikibase en los procesos operativos y asegurando una mejora sustancial en la completitud y accesibilidad de la información institucional.

1.3.2. Objetivos específicos

1. Evaluación de fuentes de datos: Realizar un análisis de repositorios públicos de información para identificar fuentes sobre publicaciones de investigadores del IMFD, evaluando criterios como completitud, disponibilidad y restricciones asociadas.

2. Desarrollo de un sistema de extracción automatizado: Implementar un sistema robusto que permita extraer, procesar y unificar información relevante de las fuentes seleccionadas, asegurando la integridad y consistencia de los datos procesados.
3. Automatización de la actualización del grafo de conocimiento: Diseñar e implementar un *pipeline* automatizado que sea capaz de poblar continuamente el grafo de conocimiento con información actualizada, garantizando la vigencia de los datos.
4. Integración con el sitio web institucional: Implementar un sistema que mantenga el sitio web del IMFD con listados actualizados de publicaciones científicas, eliminando la necesidad de intervención manual y asegurando que la información pública del Instituto refleje los datos más recientes del grafo de conocimiento.
5. Evaluación del sistema implementado: Diseñar y ejecutar experimentos para medir el impacto del sistema en la completitud y accesibilidad de la información institucional, comparando métricas antes y después de la implementación.

1.4. Evaluación

La efectividad del sistema desarrollado se evaluará mediante la implementación de dos casos de uso principales que reflejan necesidades reales del IMFD. El primer caso de uso se enfoca en validar la integración entre el grafo de conocimiento y el sitio web institucional, verificando que los datos se sincronicen correctamente y que la información mostrada públicamente sea precisa y actualizada. El segundo caso de uso se enfoca en la exportación automatizada de datos en formatos adecuados para la generación de memorias anuales, agilizando un proceso que actualmente requiere un considerable esfuerzo manual.

Para validar la calidad y completitud del sistema, se realizará una evaluación comparativa entre los datos generados automáticamente por el sistema propuesto y los datos recolectados manualmente en años anteriores. Este análisis permitirá identificar mejoras en términos de completitud y precisión de la información, así como detectar posibles áreas de mejora en los sistemas de extracción y procesamiento.

Adicionalmente, se llevará a cabo una evaluación del rendimiento técnico y la usabilidad del sistema mediante pruebas con usuarios reales. Estas pruebas involucrarán a investigadores del Instituto y a integrantes del equipo directivo, quienes representan los principales usuarios finales del sistema.

Capítulo 2

Estado del arte

El desarrollo de un sistema automatizado para la gestión de flujos de datos institucionales requiere la integración de múltiples tecnologías y enfoques de diferentes dominios de la computación. Este capítulo presenta una revisión de las tecnologías, herramientas y conceptos fundamentales que constituyen la base teórica y técnica de la solución propuesta.

2.1. Web semántica y grafos de conocimiento

La web semántica representa una evolución conceptual de la World Wide Web tradicional, transformando la visión de internet desde un conjunto de documentos enlazados hacia una red global de datos interconectados con significado explícito y estructura definida, permitiendo tanto a seres humanos como máquinas poder procesar el contenido.

Esta transformación se construye sobre una arquitectura técnica cuidadosamente diseñada y que contempla la representación de conocimiento en entornos distribuidos, como lo es la web. En la base de esta arquitectura se encuentran los URI (Uniform Resource Identifiers), que proporcionan identificadores únicos globales para cualquier concepto o entidad, esto resuelve el cómo referirse de manera inequívoca a las cosas, eliminando ambigüedades que surgirían al usar solo nombres o descripciones textuales. Sobre esta base, el estándar RDF (Resource Description Framework) [1] establece un modelo para expresar información como triples sujeto-predicado-objeto, donde cada componente se identifica mediante URIs. Esta estructura permite descomponer cualquier afirmación en unidades atómicas, que si bien parece simple, permite brindar contexto y formar relaciones entre distintos elementos. RDFS (RDF Schema) [2] extiende el poder expresivo de RDF y proporciona un vocabulario básico para definir clases, propiedades y ciertas jerarquías conceptuales entre estas, por ejemplo, decir que una propiedad es subpropiedad de otra o que alguna propiedad solo aplica a cierta clase.

Un grafo de conocimiento puede definirse como un grafo de datos diseñado para reunir y mostrar conocimiento del mundo real, en el que los nodos representan entidades y los bordes corresponden a la relación entre dichas entidades [3]. Los grafos de conocimiento son una implementación práctica de los principios de la web semántica y se caracterizan en muchos casos por utilizar ontologías para definir tipos de entidades y relaciones permitidas, empleando identificadores únicos para cada entidad y facilitando realizar consultas complejas que aprovechen la estructura interconectada de los datos. Un ejemplo importante es Wikidata [4], que centraliza los datos estructurados de plataformas como Wikipedia.

En el contexto académico e institucional, los grafos de conocimiento permiten representar relaciones complejas entre investigadores, publicaciones, proyectos, instituciones y áreas de conocimiento, facilitando análisis que van desde la identificación de colaboradores potenciales hasta el seguimiento del impacto de investigaciones específicas.

2.2. Wikibase

Wikibase es el software de código abierto que impulsa Wikidata, diseñado específicamente para crear y gestionar grafos de conocimiento estructurados. Ha ganado popularidad como herramienta para construir grafos específicos a cierta área, institución o comunidad [5]. El valor principal de Wikibase radica en su capacidad para gestionar y modelar información estructurada, especialmente en contextos donde los datos son complejos, multidimensionales y requieren representar relaciones explícitas entre entidades. Esta flexibilidad y capacidad de adaptación lo convierten en una solución ideal para estructurar grandes volúmenes de información y extraer conocimiento de manera eficiente.

El modelo de datos de Wikibase se estructura en torno a tres componentes: entidades, propiedades y declaraciones. Las entidades representan conceptos, personas, instituciones u objetos, y cuentan con un identificador único que comienza con «Q». Cada entidad puede contener múltiples declaraciones, que son afirmaciones sobre esa entidad compuestas por una propiedad y un valor. Las propiedades definen los tipos de relaciones o atributos que pueden asociarse a las entidades y ocupan identificadores que comienzan con «P». Cada declaración almacenada se estructura siguiendo el modelo RDF, traduciéndose internamente a triples de la forma sujeto-predicado-objeto: el sujeto corresponde a la entidad, el predicado a la propiedad, y el objeto al valor asociado, pudiendo ser otra entidad. Este modelo permite representar conocimiento de manera estructurada y permite editar fácilmente el grafo de conocimiento, tanto de manera manual mediante la interfaz gráfica, como de manera automatizada haciendo uso de APIs.

En la instancia de Wikibase del IMFD, este modelo se ha adaptado para representar información académica, en la que el grafo de conocimiento incluye entidades que representan investigadores, publicaciones e instituciones, vinculadas mediante propiedades diseñadas

para capturar las relaciones relevantes en el contexto académico. La Tabla 1 muestra algunas de las propiedades principales implementadas en el sistema.

PID	Propiedad	Descripción	Tipo
P2	publication date	Fecha en la que la publicación fue oficialmente lanzada	Point in time
P4	author	Individuo que participó en la creación de la publicación	Item
P5	source	Revista o conferencia en la que se publicó o presentó la publicación	Item
P11	DOI	Código único para identificar recursos digitales	External Identifier
P32	affiliation	Señala la asociación de una persona con alguna institución u organización	Item

Tabla 1: Propiedades presentes en el grafo de conocimiento del IMFD.

La Figura 1 y la Figura 2 muestran una entidad correspondiente a una publicación asociada a un investigador del IMFD (Q13) y la propiedad de afiliación (P32), respectivamente.

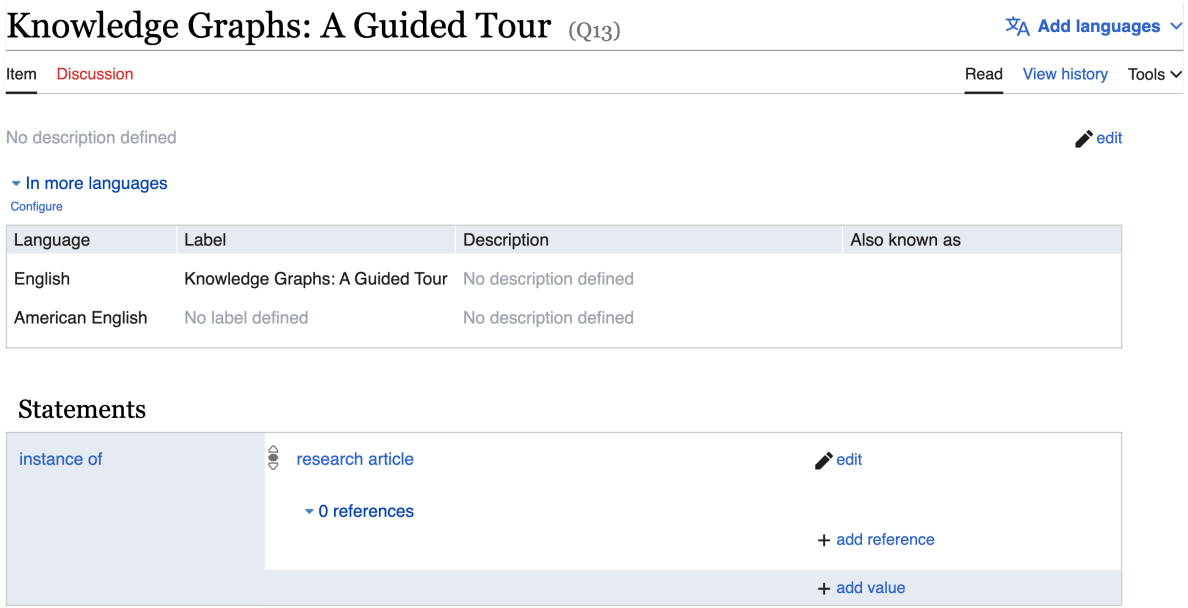

Figura 1: Interfaz de Wikibase del IMFD en la que se muestra información de una entidad.

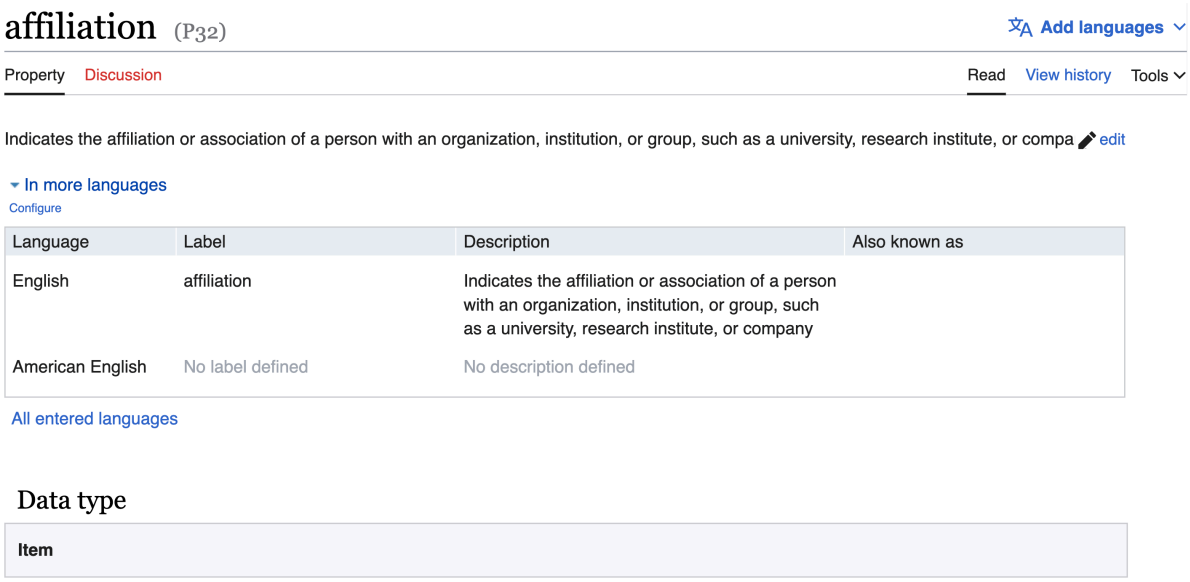

Figura 2: Interfaz de Wikibase del IMFD en la que se muestra información de una propiedad.

Para la manipulación de instancias Wikibase existen diversas alternativas. Wikibase Integrator[2] es una solución amigable desarrollada en Python que proporciona una capa de abstracción para simplificar las operaciones CRUD (crear, leer, actualizar, eliminar) sobre el grafo de conocimiento. La biblioteca permite desde consultas simples para extraer información específica hasta operaciones complejas de sincronización entre fuentes de datos y la instancia Wikibase. Esta capacidad resulta fundamental en contextos donde se quiere mantener el grafo actualizado con información que no es estática, como es el caso de la integración de metadatos académicos.

2.3. SPARQL y WDQS

SPARQL constituye el lenguaje estándar para consultar grafos de conocimiento y datos RDF, distinguiéndose fundamentalmente de SQL por su capacidad de aprovechar la estructura interconectada de los datos. Mientras las consultas SQL tradicionales operan sobre tablas relacionales y requieren múltiples operaciones JOIN para atravesar relaciones, SPARQL trabaja directamente con la representación en triples RDF que permite navegar por múltiples conexiones del grafo en una sola operación. Esta diferencia resulta especialmente poderosa para consultas complejas como encontrar todas las publicaciones de investigadores afiliados a una institución específica junto con sus coautores, donde SPARQL puede atravesar naturalmente las relaciones investigador-institución y publicación-investigador. La flexibilidad de SPARQL permite realizar agregaciones y filtros que aprovechan la semántica explícita de las relaciones en el grafo. Esta capacidad resulta especialmente

[2]https://github.com/LeMyst/WikibaseIntegrator

valiosa para extraer información estructurada que puede alimentar otros sistemas, como sitios web o herramientas de visualización.

WDQS (Wikidata Query Service) proporciona un endpoint SPARQL público que permite consultar todos los datos de Wikidata, pero no está limitado a esa base de conocimiento, sino que sirve como modelo para instancias locales de Wikibase. Actúa como un intermediario para poder obtener información desde el grafo de conocimiento mediante consultas SPARQL.

El Código 1 muestra un ejemplo de consulta SPARQL para obtener información como el nombre y el perfil de todos los investigadores afiliados al IMFD; el uso de `?person` actúa a modo de variable, de tal manera que la consulta va a devolver todas entidades que cumplen con las relaciones solicitadas. Un extracto de los resultados que devuelve la consulta anterior se puede apreciar en la Tabla 2.

```SPARQL
SELECT ?person ?personName ?profileURL
WHERE {
?person wdt:P20 wd:Q1;           # ?person es humano
        wdt:P32 wd:Q1551;        # ?person tiene afiliación con el IMFD
        wdt:P31 ?profileURL.     # ?person tiene un perfil en el sitio web

?person rdfs:label ?personName.
}
ORDER BY ?personName
```

Código 1: Ejemplo de consulta SPARQL para recuperar información de investigadores asociados al IMFD.

Person	**Person Name**	**Profile URL**
Q16	Aidan Hogan	https://imfd.cl/en/investigador/aidan-hogan/
Q190	Benjamín Bustos	https://imfd.cl/en/investigador/benjamin-bustos-c/
Q137	Bárbara Poblete	https://imfd.cl/en/investigador/barbara-poblete/
Q121	Denis Parra	https://imfd.cl/en/investigador/denis-parra/
Q316	Diego Arroyuelo	https://imfd.cl/en/investigador/diego-arroyuelo/

Tabla 2: Primeros 5 resultados de la consulta para obtener información de investigadores.

2.4. Procesos ETL y orquestación de flujos de trabajo

Los procesos ETL (Extract, Transform, Load) son fundamentales en el campo de la gestión de datos, si bien el nombre es bastante explicativo, cada etapa se encarga de:

- Extract (Extracción): Obtención de datos desde una o más fuentes heterogéneas como bases de datos relacionales, archivos de texto, APIs, servicios web, entre otros. Esta fase debe manejar diferentes formatos, protocolos para establecer una conexión con las fuentes y considerar aspectos como la frecuencia de extracción.
- Transform (Transformación): Aplicación de distintas reglas definidas para limpiar, validar y estructurar los datos extraídos. Se incluyen tareas como analizar la consistencia de los datos, llevarlos a un formato común, realizar agregaciones u otros cálculos, y aplicar ciertos estándares relacionados a la naturaleza de los datos y del negocio, como codificar información sensible de ser necesario.
- Load (Carga): Transferencia de los datos transformados al sistema de destino. Debe considerar estrategias para cargar los datos de manera incremental y completa, manejar conflictos, ya sean de disponibilidad o de fallo, y velar por el rendimiento del proceso.

Definir cómo llevar a cabo todas las tareas que contempla un proceso ETL no es trivial; la gestión automatizada de flujos de trabajo requiere herramientas especializadas. En el contexto de sistemas que manejan múltiples fuentes de datos, transformaciones sobre los mismos y dependencias entre tareas, las herramientas de orquestación modernas ofrecen ventajas significativas.

Apache Airflow[3] emerge como una solución especializada para orquestación de flujos de trabajo. La herramienta utiliza DAGs (Directed Acyclic Graphs) para definir las tareas y sus relaciones, donde cada DAG representa un flujo completo escrito en Python. Mediante estos grafos dirigidos acíclicos se establecen las dependencias entre tareas, determinando el orden de ejecución y las condiciones que deben cumplirse. La Figura 3 contiene un ejemplo de las tareas que puede contener un DAG.

Figura 3: Ejemplo simplificado de un DAG en Airflow.

Esta herramienta se destaca por algunas características como:

- Curva de aprendizaje: Al estar basado en Python, Airflow resulta bastante amigable para desarrolladores familiarizados con este lenguaje, aprovechando todo el ecosistema que provee y facilitando tanto la comprensión como el mantenimiento del código.

[3]https://airflow.apache.org/

- Interfaz web: Proporciona una interfaz que permite visualizar el estado de los flujos de trabajo en tiempo real, de esta manera se pueden identificar rápidamente procesos en ejecución, fallos y tareas pendientes.
- Operadores predefinidos: Incluye operadores listos para usar que simplifican la interacción con servicios externos, lo que reduce significativamente el tiempo de desarrollo al evitar implementar conexiones desde cero.
- Gestión de dependencias y programación: Permite establecer relaciones entre tareas definiendo qué procesos deben completarse antes que otros, esencial en *pipelines* donde la extracción precede a la transformación y a la carga. Además, utiliza expresiones cron para definir la periodicidad de ejecución de las tareas.

2.5. Fuentes de datos bibliográficos

Las fuentes bibliográficas constituyen repositorios que proporcionan metadatos sobre publicaciones académicas, sirviendo como, en este contexto, base de datos para sistemas de extracción. Algunas de estas fuentes son:

- OpenAlex [6]: Base de datos abierta que proporciona metadatos de más de 260 millones de trabajos académicos, incluyendo información sobre autores, instituciones, revistas, citas y demás.
- DBLP [7]: Bibliografía especializada en ciencias de la computación que indexa las principales revistas y conferencias de la disciplina.
- Web of Science y Scopus: Bases de datos multidisciplinarias comerciales que proporcionan información de citas, métricas de impacto, metadatos sobre autores e instituciones, y análisis más avanzados.

2.6. Literatura relacionada

El Computer Science Knowledge Graph (CS-KG) [8] es uno de los grafos de conocimiento más grandes en el ámbito de las Ciencias de la Computación, que contiene más de 350 millones de triples RDF. Su estructura comprende 41 millones de declaraciones sobre 10 millones de entidades, las cuales se encuentran vinculadas a través de 179 relaciones semánticas diferentes. Toda esta información fue extraída automáticamente de 6.7 millones de artículos científicos.

Un aspecto del CS-KG que lo hace sumamente relevante es que se actualiza de manera semestral, esto gracias a que cuenta con un *pipeline* automatizado para su construcción. Dicho *pipeline* sigue un proceso de cuatro pasos para convertir el texto de artículos científicos en un grafo de conocimiento estructurado. El primer paso utiliza cuatro herramientas diferentes para extraer información básica del texto. DyGIEpp identifica entidades junto con sus relaciones. CSO Classifier conecta el texto con temas relacionados a la computación.

OpenIE y PoS Tagger se enfocan en encontrar relaciones entre las entidades, especialmente a través de los verbos que las conectan. El segundo paso limpia y organiza la información extraída. Unifica diferentes versiones de la misma entidad, como las versiones en singular y en plural; resuelve abreviaturas y elimina información irrelevante. El tercer paso verifica que la información cumpla con las reglas de la ontología propia del CS-KG. Actúa a modo de un control de calidad que asegura que las relaciones entre entidades tengan sentido. El último paso usa aprendizaje automático para decidir qué información incluir en el grafo final. Si varios artículos mencionan la misma relación, es más probable que sea correcta. Un clasificador evalúa las relaciones menos frecuentes para decidir cuáles mantener. Finalmente, toda la información validada se convierte al formato RDF para crear el grafo de conocimiento.

Mientras que el CS-KG demuestra la construcción automatizada de grafos de conocimiento desde cero, otros enfoques aprovechan infraestructuras establecidas para crear grafos especializados, como es el caso del EU Knowledge Graph [5], que utiliza Wikibase como infraestructura para centralizar información sobre la estructura y actividades de la Unión Europea, incluyendo entidades como los países que forman parte e instituciones como el Parlamento. Sin embargo, el componente más significativo del grafo lo constituyen los proyectos y beneficiarios de programas financiados por la UE bajo la Política de Cohesión.

Similar al CS-KG, el EU Knowledge Graph implementa un sistema de mantenimiento automatizado, aunque con un enfoque diferente basado en bots especializados. El sistema incluye el Wikidata Updater Bot, que sincroniza automáticamente cambios de Wikidata, manteniendo así las entidades actualizadas. El Translator Bot se encarga de traducir entidades de un idioma a otro, proporcionando contenido multilingüe. El Geocoding Bot infiere coordenadas geográficas a partir de códigos postales. El Beneficiary Linker Bot emplea técnicas de machine learning para vincular beneficiarios mencionados como cadenas de texto con entidades existentes en Wikidata, enriqueciendo significativamente la información disponible, pudiendo generar enlaces a fuentes externas y contexto adicional.

La plataforma ofrece tres servicios principales para acceder a la información del grafo. En primer lugar, proporciona un sistema para realizar respaldos completos del grafo de conocimiento; además, cuenta con un servicio de consultas SPARQL que permite la recuperación y visualización de información de manera similar a Wikidata; finalmente, integra QAnswer como servicio de respuesta a preguntas en lenguaje natural, que permite a los usuarios realizar consultas utilizando preguntas cotidianas sin necesidad de conocimientos técnicos en SPARQL o RDF.

Capítulo 3

Extracción de datos

En este capítulo se presenta el diseño e implementación del módulo de extracción de datos de publicaciones académicas. Este proceso constituye la primera etapa del *pipeline* de datos, donde se recopila información bibliográfica desde múltiples fuentes para posteriormente ser integrada en el grafo de conocimiento del IMFD.

3.1. Fuentes de datos

La selección de fuentes bibliográficas se basó en criterios específicos para garantizar la calidad y cobertura del grafo de conocimiento, considerando los requerimientos específicos del IMFD. Los criterios incluyeron la cobertura de diversas disciplinas considerando el carácter multidisciplinario del Instituto, el acceso programático mediante APIs estables y bien documentadas, y la calidad de metadatos en términos de completitud y precisión. Adicionalmente, se consideró la compatibilidad con los recursos existentes, ya que el IMFD contaba previamente con los identificadores de sus investigadores en OpenAlex. También se evaluó la necesidad de validar la calidad editorial identificando que cierta revista está o no indexada en Web of Science y Scopus, lo que corresponde a un requerimiento institucional específico. Finalmente, se consideró la viabilidad técnica evaluando limitaciones en la cantidad de consultas y costos.

3.1.1. OpenAlex

OpenAlex se estableció como la fuente de datos principal del sistema, funcionando como el repositorio base sobre el cual se construye y enriquece la información bibliográfica. Esta elección se fundamenta en su excepcional cobertura de más de 260 millones de publicaciones académicas y más de 100 millones de autores junto a sus afiliaciones institucionales, permitiendo diferenciar mediante identificadores únicos entidades como publicaciones, autores, instituciones, journals e incluso las organizaciones que se encargan de distribuir

publicaciones. Además, OpenAlex ofrece acceso completamente gratuito sin restricciones significativas de uso, estableciendo límites de 100 mil peticiones diarias y un máximo de 10 peticiones por segundo. Adicionalmente, presenta compatibilidad directa con los OpenAlex IDs de investigadores previamente verificados por el IMFD, lo que facilita una integración más fluida. Desde el punto de vista técnico, existen librerías como PyAlex[4] que permiten interactuar con la API utilizando Python, agilizando el desarrollo y obteniendo datos estructurados en formato JSON con un esquema bien definido y conexiones explícitas entre entidades, incluyendo información completa sobre las publicaciones como el título, el Digital Object Identifier (DOI), la fecha de publicación, información de los autores y datos de la fuente de publicación, entre otros campos relevantes.

3.1.2. DBLP

DBLP representa una fuente de datos abierta y altamente especializada en ciencias de la computación, que se ha establecido como una referencia bibliográfica confiable y completa en esta área. Cuenta con información de aproximadamente 8 millones de publicaciones científicas, incluyendo artículos de revistas, tesis, libros y papers de conferencias, así como información de más de 3 millones de autores y más de 6 mil conferencias. Esta plataforma proporciona acceso a sus datos mediante una API convencional y un endpoint SPARQL del grafo de conocimiento de DBLP, el DBLP Knowledge Graph [9], que preserva todas las relaciones y datos originales de la fuente. Sus características distintivas incluyen la modelación tanto de conferencias como de journals, junto a la integración de identificadores de Wikidata, pudiendo generar una conexión con toda la información que esta base de conocimiento contiene. La principal razón de que DBLP se haya considerado una fuente secundaria de información es por su enfoque en el área de computación, dejando de lado a las otras disciplinas que son de interés para el IMFD.

3.1.3. Web of Science

Web of Science (WoS), la plataforma bibliográfica de Clarivate Analytics, constituye una de las fuentes de datos académicos más establecidas y respetadas, formando parte del estándar de referencia en la evaluación de investigaciones de alto impacto. La plataforma cuenta con herramientas para evaluar la influencia y calidad de las revistas científicas, destacando el Journal Citation Reports (JCR), que proporciona métricas como el cuartil de una revista, donde una revista en Q1 es considerada de alta calidad y relevancia, mientras que una en Q4 tiene un impacto menor. Esta métrica es particularmente importante para un centro de investigación como el IMFD, ya que permite demostrar el prestigio de los journals donde publican los investigadores asociados. Aunque Clarivate ofrece APIs para acceder a los datos, al ser una plataforma comercial requiere la compra de licencias para utilizar las herramientas avanzadas, sin embargo, también proporciona APIs con información general de publicaciones que solo requieren la creación de una cuenta para acceder a 5 mil peticiones diarias, con un máximo de 5 peticiones por segundo.

[4]https://github.com/J535D165/pyalex

3.1.4. Scopus

Scopus, la base de datos bibliográfica de la editorial Elsevier, representa otra de las fuentes de información académica más prestigiosas y ampliamente utilizadas en la evaluación de investigación, complementando la verificación de WoS al proporcionar una segunda métrica de calidad editorial que satisface completamente los criterios institucionales del IMFD. La plataforma cuenta con APIs extensamente documentadas para acceder a información sobre citaciones, métricas de investigación y otros datos bibliográficos, aunque al tratarse de una plataforma comercial requiere el pago de licencias para tener acceso completo. Sin embargo, la creación de una cuenta básica proporciona acceso a 10 mil peticiones semanales, cuota que fue posible incrementar a 35 mil peticiones semanales mediante contacto directo con ejecutivos de la organización.

3.2. Proceso de extracción

El proceso de extracción se desarrolla íntegramente en Python y contempla cuatro etapas principales, que se pueden apreciar en las tareas definidas en la Figura 4. La primera etapa consiste en extraer todas las publicaciones de los investigadores del IMFD a partir de la lista de sus OpenAlex IDs, realizando peticiones a la API que esta fuente provee mediante la librería PyAlex. La segunda etapa se encarga de consultar los mismos papers extraídos anteriormente para obtener información adicional desde DBLP; dado el contexto de esta memoria, se optó por utilizar el endpoint SPARQL mediante la librería SPARQLWrapper[5] para realizar las consultas, aprovechando la información del grafo de conocimiento de DBLP para enriquecer el grafo de conocimiento del IMFD. La tercera y cuarta etapa también parten de los papers recopilados con OpenAlex y, mediante el uso de las APIs de WoS y Scopus respectivamente, enriquecen la información con estas fuentes. Es importante mencionar que para estas dos últimas tareas es necesario incluir API keys en las requests, las cuales son proporcionadas por las organizaciones correspondientes. Todas las etapas requieren un proceso de normalización para transformar los resultados obtenidos de cada fuente a un formato común, facilitando así el procesamiento posterior.

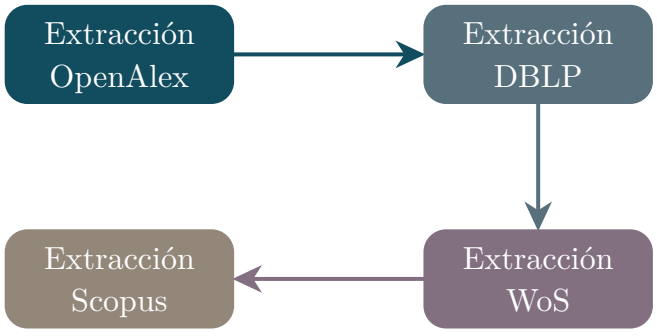

Figura 4: Flujo de extracción de datos.

[5]https://github.com/RDFLib/sparqlwrapper

3.2.1. Atributos considerados

Desde un comienzo, el IMFD puso a disposición las memorias anuales con información que abarca las distintas áreas del Instituto, incluyendo actividades científicas, publicaciones, presentaciones en congresos y demás. En específico, la información sobre las publicaciones fue de especial ayuda para definir los atributos que se quieren obtener de las distintas fuentes a considerar, después de revisar con sumo cuidado estos documentos se establecieron todos los atributos a extraer, los cuales se presentan en la Tabla 3.

Fuente	Campo	Descripción
OpenAlex	title	Nombre asociado a una publicación.
	id	Identificador de OpenAlex para una publicación
	doi	Código utlizado para identificar de manera única objetos digitales.
	publication_date	Fecha de publicación del paper.
	updated_date	Última fecha de actualización de la información del paper.
	authorships	Información sobre los autores de la publicación, incluyendo nombres, afiliaciones y ORCID.[6]
	type	Tipo de publicación.
	primary_location	Principal fuente en la que se encuentra publicado el paper, incluye nombre y ISSN.[7]
	biblio	Información bibliográfica.
	abstract_inverted_index	Abstract de la publicación.
DBLP	published_in_stream	Serie de conferencias o journals en que el paper fue publicado.
	stream_title	Nombre de la conferencia o journal.
	year_of_event	Año en que se llevó a cabo la conferencia.
	wikidata_id	QID de Wikidata para el paper.
	isbn	Identificador único para libros.
WoS	wos_indexed	Indica si una publicación está indexada en WoS o no.
	wos_id	Identificador del paper según WoS.
Scopus	scopus_indexed	Indica si una publicación está indexada en Scopus o no.
	scopus_id	Identificador del paper según Scopus.

Tabla 3: Información extraída de las distintas fuentes de datos.

[6]Identificador único para científicos y autores académicos.

[7]Identificador único de publicaciones serializadas, como revistas.

3.2.2. Extracción desde OpenAlex

Como se mencionó anteriormente, el sistema utiliza la librería de Python PyAlex para realizar consultas a la API de OpenAlex, utilizando los OpenAlex IDs de los investigadores del IMFD como punto de partida para obtener todas las publicaciones de dichos autores. Es importante destacar que cada autor puede contar con más de un OpenAlex ID, por lo que el sistema está diseñado para iterar por todos identificadores asociados a los investigadores. En un comienzo se quiso utilizar el nombre de la persona para obtener la información de sus publicaciones, pero no resultó una estrategia viable por problemas de ambigüedad, ya que múltiples autores pueden compartir el mismo nombre o variaciones similares.

El proceso de consulta está optimizado mediante un sistema de paginación que extrae 200 resultados por página, minimizando el número de requests necesarios y reduciendo la carga en la API. De esta forma, si un investigador tiene menos de 200 papers publicados, solo se estaría haciendo un llamado a la API de OpenAlex. En el mismo sentido, para respetar los límites de la API, se implementa un mecanismo de control que introduce pausas programadas cada 10 requests, permitiendo que el sistema sea sostenible para extracciones largas. Además, la estrategia de consulta permite extraer únicamente los campos necesarios para el procesamiento posterior, lo que reduce significativamente el tiempo de respuesta y la carga sobre OpenAlex, ya que para cada publicación no se solicitan los cerca de 50 atributos disponibles en la fuente, sino que solamente 10 de ellos.

3.2.3. Enriquecimiento con fuentes secundarias

Una vez completado el proceso de extracción desde OpenAlex, el sistema procede a enriquecer la información de las publicaciones mediante la consulta a las otras tres fuentes de datos, DBLP, Web of Science y Scopus. Aunque cada publicación extraída de OpenAlex incluye su identificador único específico de esa plataforma, estos identificadores no son transferibles ni reconocidos por otras bases de datos académicas, lo que imposibilita su uso para la identificación cruzada entre fuentes. Por esta razón, el proceso de enriquecimiento ocupa el DOI como llave de identificación universal.

La integración con DBLP utiliza consultas al endpoint SPARQL de DBLP para obtener información especializada sobre publicaciones en ciencias de la computación. Para cada publicación, el sistema construye dinámicamente una consulta SPARQL que especifica todos los campos de interés, ejecutándola posteriormente con SPARQLWrapper para obtener los resultados correspondientes. El Código 2 muestra un ejemplo de esta consulta, donde se puede observar el uso del prefijo `dblp:` para referenciar el esquema RDF de DBLP y cómo se utiliza el DOI de la publicación para identificarla en la base de datos, extrayendo todos los campos de interés. Las consultas SPARQL están diseñadas para extraer campos que pueden tener más de un valor asociado, utilizando `GROUP_CONCAT` para juntar múltiples valores en uno solo. Esto último ocurre en específico para el ISBN, ya que si bien corresponde a un identificador para libros, puede ocurrir que un mismo libro tenga

diferentes formatos de distribución, como una versión impresa y una versión digital, cada uno con su identificador propio.

```SPARQL
PREFIX dblp: <https://dblp.org/rdf/schema#>
SELECT ?publ ?publishedInStream ?streamTitle ?yearOfEvent ?wikidata
  (GROUP_CONCAT(DISTINCT ?isbn; separator='|') as ?isbn_concat)
WHERE {
  ?publ dblp:doi <https://doi.org/10.1145/3447772> .
  OPTIONAL { ?publ dblp:publishedInStream ?publishedInStream . }
  OPTIONAL { ?publ dblp:streamTitle ?streamTitle . }
  OPTIONAL { ?publ dblp:yearOfEvent ?yearOfEvent . }
  OPTIONAL { ?publ dblp:wikidata ?wikidata . }
  OPTIONAL { ?publ dblp:isbn ?isbn . }
}
GROUP BY ?publ ?publishedInStream ?streamTitle ?yearOfEvent ?wikidata
```

Código 2: Consulta SPARQL para obtener información de un artículo desde DBLP.

La verificación en Web of Science se realiza mediante una solicitud HTTP GET a la API de Clarivate, utilizando una estrategia de consulta directa que emplea el DOI como parámetro de búsqueda. La presencia de la publicación se determina evaluando si la respuesta de la API contiene un identificador único de WoS en los metadatos del documento encontrado, cuando esto ocurre se considera que el paper fue publicado en una revista indexada en WoS. La verificación en Scopus sigue una metodología análoga a la implementada para Web of Science, empleando la API de Elsevier.

3.2.4. Transformaciones sobre los datos

Procesamiento de autores

El procesamiento de información de autores contempla múltiples transformaciones para normalizar y estructurar los datos del campo `authorships`. El sistema realiza una limpieza de los identificadores de OpenAlex, eliminando los prefijos URL completos para mantener únicamente el identificador esencial. Para los identificadores ORCID, se aplica una normalización similar que extrae el identificador limpio sin la URL completa. La información institucional se procesa para obtener la institución primaria de cada autor, incluyendo tanto el identificador de OpenAlex como el nombre de la institución.

Procesamiento de fuentes de publicación

El procesamiento de fuentes de publicación se enfoca específicamente en extraer y estructurar la información del campo `primary_location` sobre la fuente donde fue publicado el artículo y su organización anfitriona. El sistema procesa únicamente estos dos niveles de información: la fuente directa de publicación (revista, conferencia, libro, etc.) y la

organización que aloja o publica dicha fuente. Durante este procesamiento, al igual que antes, se normalizan los identificadores de OpenAlex mediante la eliminación de prefijos.

Procesamiento bibliográfico

El procesamiento de datos bibliográficos se encarga de unificar la información de paginación contenida en el campo `biblio`. El sistema combina la información de la primera y última página de la revista en que fue publicado el artículo, creando un formato «primera--última». En casos donde solo está disponible una página, se mantiene como página única, asegurando que no se pierda información valiosa.

Procesamiento de abstracts

El procesamiento de abstracts desde el campo `abstract_inverted_index` convierte el formato en que OpenAlex entrega el abstracts a texto plano. OpenAlex, debido a restricciones legales, proporciona los resúmenes en formato de índice invertido donde cada palabra está asociada con sus posiciones en el texto original. El algoritmo de reconstrucción crea un arreglo de palabras basado en las posiciones máximas encontradas en `abstract_inverted_index` y luego coloca cada palabra en su posición correcta para reconstruir el texto completo.

Procesamiento de fuentes secundarias

El procesamiento para DBLP, WoS y Scopus se basa principalmente en limpiar los identificadores que proveen, eliminando prefijos innecesarios para obtener únicamente los códigos. Para DBLP se procesa además el valor obtenido para el ISBN, que puede incluir múltiples códigos, separándolos en arreglos individuales.

La Tabla 4 contiene algunos ejemplos de cómo viene la información directamente desde las fuentes de información y cómo lucen luego de aplicar las transformaciones descritas previamente.

Propiedad	Formato Original	Formato Transformado
ID de OpenAlex	https://openalex.org/W3010336026	W3010336026
ORCID del autor	https://orcid.org/0000-0001-9482-1982	0000-0001-9482-1982
ID institución	https://openalex.org/I69737025	I69737025
DOI	https://doi.org/10.1145/3447772	10.1145/3447772
Wikidata ID	http://wikidata.org/entity/Q113482045	Q113482045
WoS ID	WOS:000670593100011	000670593100011
Scopus ID	SCOPUS_ID:85108965182	85108965182

Tabla 4: Transformaciones de limpieza de identificadores.

3.2.5. Características del sistema

Consolidación de la información

Como resultado del procesamiento, se obtiene una estructura JSON unificada que combina de manera coherente todos los datos extraídos de las diferentes fuentes, creando un formato estándar que facilita la integración posterior. La estructura mantiene la información de OpenAlex como base, complementándola con datos específicos de cada fuente secundaria en secciones claramente definidas. Aunque el DOI se utiliza como identificador principal para el enriquecimiento, desde OpenAlex también se recuperan publicaciones que no cuentan con un DOI, las cuales se mantienen separadas y se excluyen del enriquecimiento con fuentes secundarias dado la imposibilidad de garantizar que la información obtenida corresponda al mismo artículo.

El DOI no solo se utiliza para identificar una publicación en distintas fuentes de información, sino que también es útil para evitar duplicados con la información obtenida desde OpenAlex, manteniendo una sola entrada por DOI. Esta aproximación es especialmente importante cuando se procesan múltiples autores que pueden haber colaborado en las mismas publicaciones.

Manejo de errores

El sistema implementa un manejo de errores robusto mediante bloques try-except de Python que capturan y procesan las excepciones de manera controlada, evitando que los errores se propaguen y comprometan la ejecución del proceso completo. Esta arquitectura permite la recuperación automática ante fallos individuales, asegurando que el sistema continúe funcionando incluso cuando artículos específicos resultan problemáticos. Adicionalmente, el sistema tiene un enfoque permisivo con respecto a los campos de información, asignando valores nulos a aquellos atributos para los cuales no se encuentra información disponible en lugar de detener la ejecución, garantizando así que la ausencia de datos específicos no comprometa la obtención del resto de la información. Cabe mencionar que se cuenta con un logging detallado que orienta sobre qué está haciendo el sistema de extracción en cada momento, proporcionando un seguimiento completo del proceso y dando visibilidad a los problemas que pueden ocurrir durante la ejecución.

En la fase de pruebas se identificaron problemas de intermitencia en la plataforma de DBLP, lo que hacía que el proceso tomara mucho más tiempo del que se espera en condiciones normales, ya que en cada iteración se excedía el tiempo máximo de espera. Ante esto, se implementó un timeout configurable de 30 segundos por defecto, que deshabilita las consultas a DBLP si se detectan problemas de conectividad; este enfoque permite que el sistema continúe funcionando incluso cuando la fuente experimenta problemas de disponibilidad.

Actualización Incremental

Se implementa un mecanismo de actualización incremental que utiliza la fecha de la última ejecución exitosa y la compara con los timestamps de `updated_date` disponibles en OpenAlex para optimizar el procesamiento en ejecuciones posteriores. Este enfoque permite procesar únicamente las publicaciones que han sido modificadas o agregadas desde la última ejecución, reduciendo significativamente el tiempo de procesamiento y la carga en las APIs externas, ya que se evita consultar DBLP, Web of Science y Scopus para publicaciones que no han experimentado cambios.

Al utilizar OpenAlex como fuente principal para determinar qué publicaciones requieren actualización, se establece un criterio que dictamina cuáles publicaciones serán posteriormente actualizadas en el grafo de conocimiento del IMFD. Si bien esta decisión de diseño implica que las actualizaciones dependen exclusivamente de los cambios reportados por OpenAlex, no trae mayores repercusiones dado que esta fuente mantiene un registro actualizado de la literatura académica. Esta optimización es especialmente valiosa para sistemas que se van a estar ejecutando periódicamente y se quiere evitar tener que reprocesar todo el conjunto de datos.

3.3. Integración con Airflow

Hasta ahora, el proceso de extracción se ha descrito como una sola iteración sobre todas las publicaciones de los investigadores del IMFD, pero la implementación real requiere automatización y orquestación para evitar la intervención manual. En este contexto, Airflow toma un papel fundamental como orquestador de procesos, aprovechando que el IMFD ya cuenta con un servidor que posee una instancia de Airflow funcionando.

La integración con Airflow permite acceder a variables del sistema como la hora de inicio y finalización de la última ejecución exitosa, siendo particularmente relevante la primera para el mecanismo de actualización incremental. Al utilizar la hora de inicio de la última ejecución exitosa para compararla con el `updated_date` de OpenAlex, se garantiza que ninguna publicación actualizada durante el proceso anterior quede excluida del procesamiento, abordando así el caso excepcional donde una publicación se actualice mientras el sistema está en ejecución.

Desde Airflow también se configuran aspectos operacionales como el número de reintentos ante fallos, la demora entre intentos y la periodicidad de ejecución. Se estableció una frecuencia semanal que busca un equilibrio entre la frecuencia con que los investigadores publican nuevos trabajos y las actualizaciones que se van haciendo sobre artículos ya publicados.

La implementación en Airflow consta de dos tareas principales, como se muestra en la Figura 5. La primera tarea se encarga de obtener los identificadores de los investigadores del IMFD, un proceso que inicialmente recuperaba los IDs desde un archivo almacenado en un bucket de Amazon S3, pero que posteriormente se modificó por un enfoque más dinámico que se detalla en el Sección 4.1. La segunda tarea ejecuta el proceso de extracción propiamente tal, utilizando los identificadores obtenidos en la etapa anterior para realizar las consultas a OpenAlex y el posterior enriquecimiento con las fuentes secundarias.

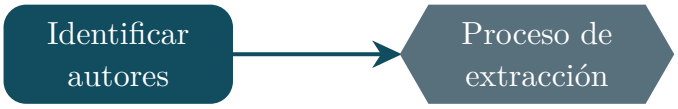

Figura 5: Tareas definidas en Airflow para extraer la información.

Capítulo 4

Integración con Wikibase

La integración con Wikibase constituye uno de los componentes centrales del sistema desarrollado, permitiendo la estructuración de la información bibliográfica recolectada de los investigadores del IMFD. Este capítulo describe la implementación de un sistema robusto que transforma los datos obtenidos de diversas fuentes bibliográficas para incluir esta nueva información en el grafo de conocimiento del Instituto[8].

4.1. Identificar autores

En un inicio se estaba ocupando una lista con los OpenAlex IDs de los investigadores y a partir de esa lista se obtenían metadatos de los artículos publicados, pero este enfoque ignoraba completamente que en el grafo de conocimiento del IMFD ya se contaba con perfiles para los investigadores, por lo que se decidió añadir esta información al grafo y complementar los perfiles de cada investigador. Esto se hizo mediante un script en Python que automatiza el proceso de carga de OpenAlex IDs directamente en el grafo de conocimiento institucional. Para ello se utiliza la librería WikibaseIntegrator, estableciendo primero la conexión con el grafo a través de las credendiales de usuario y contraseña previamente dadas por el IMFD. La implementación requirió la creación previa de la propiedad OpenAlex ID directamente desde la interfaz gráfica de Wikibase, configurándola como External ID para que así pueda existir una conexión directa entre la entidad en el grafo y la entidad en OpenAlex. Es importante destacar que esto tiene como prerequisito que el equipo del IMFD agregue manualmente cada nuevo investigador al grafo de conocimiento junto con su OpenAlex ID para que sea considerado por el sistema.

El proceso comienza leyendo los datos desde un bucket de S3 que contiene un archivo JSON con la asociación entre nombres de investigadores y sus respectivos OpenAlex IDs.

[8]https://wikibase.imfd.cl

Para cada investigador, el script ejecuta una búsqueda de entidades en el grafo utilizando el nombre del investigador para localizar su perfil existente y recuperar su QID. Una vez identificada la entidad correspondiente, toma cada identificador y crea una declaración de tipo External ID utilizando la propiedad OpenAlex ID previamente configurada en el grafo. Es importante mencionar que el comportamiento por defecto de WikibaseIntegrator al intentar añadir un claim es reemplazar el contenido si ya existía una declaración para esa propiedad, o crear una nueva si no existe. Por esta razón, el script utiliza específicamente la acción APPEND_OR_REPLACE, que compara los claims existentes en el grafo y únicamente añade aquellos que no se encontraban previamente, permitiendo así mantener todos los OpenAlex IDs asociados a cada investigador sin crear duplicados, lo que evita que en futuras ejecuciones se vuelvan a añadir los mismos identificadores al perfil del autor.

Para recuperar los OpenAlex IDs de los investigadores desde el grafo se utiliza una consulta SPARQL, ejecutándola mediante SPARQLWrapper directamente sobre el endpoint que la instancia de Wikibase provee para ello[9]. Esta consulta es la que se ejecuta semanalmente para dar marcha al flujo de extracción y se puede observar en el Código 3.

```SPARQL
SELECT ?person ?personName ?openAlexID
WHERE {
?person wdt:P20 wd:Q1;           # ?person es humano
        wdt:P32 wd:Q1551;        # ?person tiene afiliación con el IMFD
        wdt:P37 ?openAlexID.     # ?person tiene al menos un OpenAlex ID

?person rdfs:label ?personName.
}
ORDER BY ?personName
```

Código 3: Consulta SPARQL para obtener los OpenAlex IDs de los autores.

4.2. Cargar las publicaciones en el grafo de conocimiento

4.2.1. Entidades y propiedades

Un aspecto de la integración es el mapeo entre los campos de datos bibliográficos extraídos y las propiedades del esquema del grafo de conocimiento del IMFD. Un diccionario de Python establece asociaciones explícitas entre campos extraídos como título, DOI, fecha de publicación y autores con sus respectivos identificadores de propiedad en Wikibase. Fue necesario crear nuevas propiedades en el grafo para representar completamente la información bibliográfica extraída de las fuentes, tales como «publisher» para la organización responsable de distribuir una revista y «wikidata_id» para el identificador Q de Wikidata, entre otras propiedades. De manera similar, otro diccionario define las correspondencias

[9]https://query.wikibase.imfd.cl

con entidades previamente creadas en el grafo, incluyendo seres humanos para representar autores (Q1), artículos de investigación (Q2), categorías de indexación como Scopus (Q4) y Web of Science (Q18). Este mapeo bidireccional entre los datos extraídos y la estructura del grafo facilita la inserción de nuevas publicaciones, junto a todos los statements que nutren de información a la publicación. La lista completa con las propiedades preexistentes y creadas en el grafo se encuentra en la Tabla 5.

Propiedades Preexistentes			**Propiedades Creadas**		
PID	**Propiedad**	**Tipo**	**PID**	**Propiedad**	**Tipo**
P2	Publication date	Point in Time	P37	OpenAlex ID	External ID
P3	Publication category	Item	P38	Publisher	Item
P4	Authored by	Item	P39	ORCID	External ID
P5	Source	Item	P40	OpenAlex type	String
P6	Volume	String	P42	Last updated	Point in Time
P7	Issue	String	P43	Abstract	Monolingual Text
P8	Title	String	P44	Wikidata ID	External ID
P9	Pages	String	P45	Published in series	String
P10	ISSN	External ID	P46	Series title	String
P11	DOI	String	P47	Year of event	Point in Time
P19	Ordinal value	Quantity	P48	ISBN	External ID
P20	Instance of	Item	P49	Scopus ID	External ID
P32	Affiliation	Item	P50	WoS ID	External ID

Tabla 5: Propiedades preexistentes y creadas en Wikibase.

4.2.2. Publicaciones existentes en el grafo

En la instancia de Wikibase ya existían registros de publicaciones académicas; estos fueron creados a partir de las memorias anuales que realiza el Instituto con información validada por directivos e investigadores del IMFD. Ante esto, el sistema implementa una estrategia para identificar publicaciones ya existentes en Wikibase, utilizando el DOI como identificador único. Este análisis ejecuta una consulta SPARQL que recupera todos los DOIs previamente almacenados, permitiendo clasificar las publicaciones entrantes en dos categorías: aquellas que requieren actualización y aquellas que necesitan crearse desde cero. La consulta que se ocupa es la que se aprecia en el Código 4 y está diseñada para identificar entidades que son instancias de artículos de investigación y poseen un DOI asociado. Este enfoque es particularmente útil considerando que se trata de un proceso que se ejecuta de manera periódica, por lo que en distintas ejecuciones se pueden crear publicaciones que ya existen en Wikibase. Sin hacer este chequeo previo se corre el riesgo de duplicar la información en el grafo de conocimiento, teniendo más de una instancia para la misma

publicación. La verificación de existencia también optimiza el rendimiento del sistema al evitar operaciones de escritura innecesarias en Wikibase, reduciendo la carga sobre el servicio.

```sparql
SELECT ?publication ?doi
WHERE {
  ?publication wdt:P20 wd:Q2;    # ?publication es un paper de investigación
               wdt:P11 ?doi .    # ?publication tiene DOI
}
```

Código 4: Consulta SPARQL para obtener los papers existentes en el grafo de conocimiento.

4.2.3. Dependencias y relaciones entre entidades

Si bien sería relativamente sencillo crear publicaciones en Wikibase dejando todos los campos como texto plano, esto no aprovecha las capacidades semánticas que provee la plataforma, como establecer enlaces directos entre dos entidades, eliminando la posibilidad de navegar relacionalmente por el grafo. También se pierden las validaciones automáticas que realiza Wikibase sobre los tipos de datos, permitiendo inconsistencias y errores que comprometen la integridad de los datos. Ahora bien, crear una publicación en Wikibase con información de sus autores, requiere que dichos autores existan en la plataforma para poder efectivamente asociarlos a la publicación, por lo que existe una dependencia en las entidades del grafo de conocimiento. Esto último no solo ocurre para los autores, sino que también para las instituciones a la que están afiliados los autores, la fuente en que se publicó el artículo y la organización editora.

Ante esto, el sistema es capaz de gestionar automáticamente la creación de entidades relacionadas antes de procesar las publicaciones principales, identificando todas las entidades referenciadas en una publicación específica. La estrategia de creación sigue un orden específico: primero se procesan organizaciones anfitrionas e instituciones, seguidas por fuentes de publicación y finalmente autores. Este orden jerárquico asegura que todas las referencias requeridas estén disponibles, dado que la fuente puede tener una organización anfitriona y cada autor puede tener una afiliación. Es fundamental evitar que por cada paper se creen todas las entidades que necesita sin considerar que pueden existir previamente en el grafo, no solo por coincidencia, sino porque es común que investigadores tengan más de alguna publicación en conjunto y también que la afiliación de un investigador sea la misma para un conjunto de publicaciones. Para evitar esto, se planteó ocupar otra consulta SPARQL para comprobar la existencia o no de las entidades en el grafo, ocupando el OpenAlex ID como llave primaria, pero no sería una sola consulta por artículo, sino que sería una consulta por cada autor y otras tres para el resto de atributos, aumentando significativamente la carga sobre el servicio de consultas. Afortunadamente, SPARQL cuenta con la cláusula `VALUES` para consultar por múltiples valores específicos de manera eficiente con una sola operación. El Código 5 muestra un ejemplo de estas consultas, donde se buscan entidades

que tengan ciertos OpenAlex IDs. De existir, se recuperan sus QIDs, mientras que para aquellas que no existen se procede a crear nuevas entidades mediante WikibaseIntegrator. Durante la creación de nuevas entidades, el sistema incluye claims con toda la información bibliográfica recolectada, como el ISSN para las fuentes de publicación, identificadores de OpenAlex (vitales para reconocerlas en futuras iteraciones), y metadatos generales como que cada autor es una instancia de ser humano. Cabe mencionar que se lleva un diccionario con todas las entidades que se crearon como las que no, teniendo los OpenAlex IDs como llave y los QIDs como valor.

```SPARQL
SELECT ?entity ?openAlexID
WHERE {
  ?entity wdt:P37 ?openAlexID .    # ?entity tiene un OpenAlex ID
  VALUES ?openAlexID {
    "A5070504151"                  # ?openAlexID puede ser "A5070504151"
    "A5009309241"                  # ?openAlexID puede ser "A5009309241"
    "A5073103554"                  # ?openAlexID puede ser "A5073103554"
  }
}
```

Código 5: Consulta SPARQL ocupando `VALUES`.

No basta con solo crear las entidades en Wikibase, también es necesario incluir todas las relaciones semánticas entre ellas. Por ello, el sistema itera sobre todas las entidades previamente creadas o identificadas, utilizando el mapeo de identificadores OpenAlex a QIDs de Wikibase, recuperando el ítem desde el grafo y procede a agregar claims relacionales específicos según el tipo de entidad. Para entidades como revistas o journals se verifica si existe información sobre la organización responsable de su publicación; si esta organización ya existe en el grafo como una entidad mapeada, se establece una relación «publisher» utilizando la propiedad correspondiente. Para autores, se procesa la información de afiliación institucional cuando está disponible. Si la institución existe en el grafo, se establece una relación de afiliación utilizando la propiedad «affiliation» en Wikibase. Este proceso de formar las relaciones se ejecuta en cada iteración del sistema, incluso cuando no se han creado nuevas entidades, ya que las fuentes bibliográficas actualizan su información y datos que no estaban disponibles en extracciones anteriores pueden aparecer en ejecuciones posteriores. Por ejemplo, un autor que inicialmente no tenía información de afiliación en OpenAlex puede tenerla disponible en una actualización subsecuente, permitiendo establecer relaciones institucionales que antes no existían, enriqueciendo progresivamente la completitud del grafo de conocimiento.

La Figura 6 ilustra la secuencia de tareas implementada para manejar las dependencias y relaciones en el grafo de conocimiento. El flujo comienza con la consulta de identificadores existentes para evitar duplicación, continúa con la creación de nuevas entidades cuando

es necesario, y culmina con el establecimiento de relaciones semánticas entre todas las entidades involucradas.

Figura 6: Tareas para manejar dependencias y relaciones en el grafo de conocimiento.

4.2.4. Creación y actualización de publicaciones

Sabiendo que todas las entidades necesarias que referencia una publicación ya se encuentran en Wikibase y que se conoce si una publicación ya existe en el grafo desde antes, solo queda crear la entidad para el artículo de investigación cuando sea necesario, incluyendo toda la información bibliográfica recolectada previamente como statements estructurados en Wikibase. Para esto justamente es que se separó la información de las publicaciones en dos grupos: las que deben crearse por primera vez y las que deben actualizarse. Además, se mantiene en un diccionario los QIDs de todas las entidades que referencia cada publicación, facilitando la construcción de las relaciones semánticas correspondientes.

Para llevar la información resultante del proceso de extracción al formato de claims que maneja WikibaseIntegrator, se implementó un método que recibe toda la información de una publicación y para cada campo en el que se tiene un valor se crea un claim con el tipo de dato y propiedad adecuada. Aquí toma relevancia la asociación bidireccional entre el nombre del atributo extraído y su correspondiente PID en el grafo, ya que el claim va a identificar la propiedad que se está ocupando según el PID que se indique. Los tipos de datos también requieren especial cuidado en su configuración. Los claims de tipo Point in Time, por ejemplo, deben especificar una precisión temporal según el contexto de la información, mientras que «year_of_event» para conferencias académicas utiliza precisión a nivel año, «publication_date» ocupa precisión a nivel día para capturar la fecha exacta de publicación. De manera similar, los claims de tipo Item requieren particular atención en cuanto a sus valores, ya que no reciben directamente el nombre del autor o el título de la fuente que publicó el artículo, sino que deben referenciar el QID de la entidad que ya existe en el grafo de conocimiento. Los autores de una publicación requieren un tratamiento adicional debido a que el orden en que aparecen listados tiene significado académico. Para preservar esta información, cada claim de autoría incluye un qualifier de tipo «ordinal_value» que especifica la posición numérica del autor, añadiendo contexto al claim original. Este enfoque basado en identificadores únicos es esencial para mantener la integridad entre entidades interconectadas. Como resultado de este método, se obtiene una lista completa de claims que representan todos los aspectos bibliográficos y relacionales de la publicación.

El sistema trata de forma similar la creación de nuevas publicaciones en el grafo y la actualización de artículos ya existentes, haciendo una diferencia cuando se comienza a

procesar la información de cada publicación. Para publicaciones existentes, se utiliza el QID previamente identificado durante el análisis de DOIs para recuperar la entidad desde Wikibase. En contraste, para nuevas publicaciones se instancia una entidad completamente nueva, generando automáticamente un QID único que identificará la publicación en el grafo. Una vez obtenida la entidad, el procesamiento posterior es idéntico, se agregan todos los claims de la lista generada anteriormente utilizando la configuración `APPEND_OR_REPLACE`, que permite la inserción de nueva información sin generar duplicados y que resulta particularmente útil para propiedades que pueden tener más de un claim asociado, como ocurre con los autores. Hasta este punto, todos los cambios introducidos se han realizado sobre la representación local de la entidad, por lo que el paso final consiste en sincronizar estos cambios con la instancia de Wikibase, completando el ciclo de integración de datos bibliográficos al grafo de conocimiento.

Una característica crucial del sistema es su naturaleza idempotente, lo que significa que ejecutar el proceso múltiples veces con los mismos datos produce el mismo resultado sin crear duplicados o inconsistencias. Esto se logra principalmente mediante la verificación previa de existencia de DOIs para publicaciones y el uso de identificadores OpenAlex como claves primarias para entidades. Complementando esta característica, el sistema implementa una estrategia robusta de manejo de errores que garantiza que problemas con publicaciones individuales no comprometan la ejecución completa del proceso. Cada operación crítica está envuelta en bloques try-except que capturan el error y registran lo ocurrido en los logs.

4.3. Flujo en Airflow

En el capítulo anterior se abordó el proceso hasta la extracción y enriquecimiento de la información bibliográfica, donde el flujo en Airflow culminaba con la generación de archivos JSON que contenían los datos procesados de las publicaciones. El flujo completo implementado actualmente, como se ilustra en la Figura 7, incorpora la integración con Wikibase como una tercera tarea que se ejecuta inmediatamente después de completarse la extracción de datos. Esta nueva etapa toma los datos bibliográficos procesados de la fase anterior y los inyecta en el grafo de conocimiento del IMFD como statements estructurados. La cuarta y última tarea del flujo corresponde a una estrategia de gestión de recursos específica de Airflow. Dado que no es directo utilizar el valor retornado por una tarea en otra subsecuente, Airflow almacena automáticamente los outputs de los operadores en XCom, un mecanismo interno que permite el intercambio de datos entre tareas dentro de un DAG. A largo plazo, tras múltiples ejecuciones del flujo, este almacenamiento puede volverse considerable en términos de memoria. Por esta razón, al final del proceso se liberan explícitamente las variables almacenadas en XCom, específicamente la variable que contiene los OpenAlex IDs de los autores y toda la información extraída de las fuentes bibliográficas. Es importante destacar que aunque estos datos se eliminen de XCom, ya han sido persistidos en Wikibase durante la tercera tarea, donde quedan disponibles para ser

consultados en futuras ejecuciones del flujo, asegurando así un uso eficiente de los recursos del sistema.

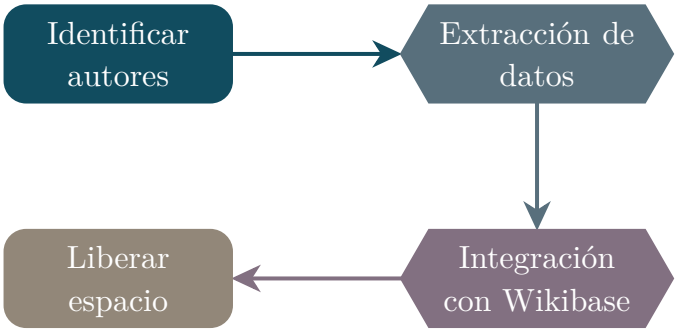

Figura 7: Flujo completo en Airflow.

Capítulo 5

Evaluación

La evaluación del sistema implementado constituye una etapa crucial para demostrar el cumplimiento de los objetivos planteados y validar la efectividad de la solución desarrollada. Este capítulo presenta una evaluación comprehensiva que aborda múltiples dimensiones del impacto del sistema: la accesibilidad de la información a través de su integración con el sitio web institucional, la evaluación de la completitud del sistema mediante la comparación entre memorias anuales históricas y reportes generados automáticamente a partir del grafo de conocimiento, y la usabilidad de las herramientas de gestión de la información bibliográfica.

5.1. Plugin sitio web

El sitio web del IMFD está hecho con WordPress; la forma que tiene este gestor de contenidos para interactuar con servicios externos, como el grafo de conocimiento del Instituto, es mediante plugins. Ante esto, para poblar el sitio web institucional con información de las publicaciones de los investigadores, es que se implementó un plugin ocupando el lenguaje PHP para poder ejecutar una consulta SPARQL sobre Wikibase que obtiene atributos de interés sobre las publicaciones, como el título, la fecha de publicación, los autores, la fuente y su OpenAlex ID. La consulta está diseñada para reflejar la estructura de presentación deseada en el sitio web, donde las publicaciones se organizan por año y se muestran en orden cronológico descendente, posicionando las más recientes en primer lugar. En el Código 6 se presenta un ejemplo de una versión simplificada de esta consulta para las publicaciones del año 2025, donde se puede observar el manejo de propiedades con múltiples valores asociados, los nombres de autores se agrupan separándolos con comas («,»), mientras que las categorías de publicación se delimitan con punto y coma («;»). La consulta utiliza la propiedad `rdfs:label` para obtener los nombres de las entidades, de lo contrario se mostrarían sus identificadores internos de Wikibase.

```sparql
SELECT ?article ?title ?publicationDate
 (GROUP_CONCAT(DISTINCT ?categoryLabel; SEPARATOR="; ") AS ?pubCategories)
 (GROUP_CONCAT(DISTINCT ?authorLabel; SEPARATOR=", ") AS ?authors)
WHERE {
  ?article wdt:P20 wd:Q2 .                    # artículos de investigación
  ?article wdt:P8 ?title .
  ?article wdt:P2 ?publicationDate .
  FILTER(YEAR(?publicationDate) = 2025)       # filtrar por año
  OPTIONAL {
    ?article wdt:P3 ?category .
    ?category rdfs:label ?categoryLabel .     # nombre de la categoría
  }
  OPTIONAL {
    ?article wdt:P4 ?author .
    ?author rdfs:label ?authorLabel .         # nombre del autor/a
  }
GROUP BY ?article ?title ?publicationDate
ORDER BY DESC(?publicationDate)               # orden descendente
```

Código 6: Consulta SPARQL simplificada para obtener información de artículos publicados el año 2025.

El plugin además se encarga de procesar los resultados de la consulta y estructurar la información de cada publicación mediante tarjetas individuales, mostrando información acotada que incluye elementos como el título y los autores, y que al ser seleccionadas despliegan todos los datos obtenidos, como el abstract y si está indexada en WoS o en Scopus. Cada tarjeta incluye dos redirecciones, una a la página oficial de la publicación mediante el DOI y otra a la página de la publicación en OpenAlex. Adicionalmente, se presenta un breve resumen de la cantidad total de publicaciones encontradas. El plugin muestra únicamente los campos para los cuales existe información disponible. Esta aproximación evita elementos vacíos en la interfaz, garantizando que publicaciones sin abstract u otros metadatos opcionales mantengan una presentación coherente.

El filtrado por año de publicación permite organizar el contenido cronológicamente, facilitando la creación de secciones desplegables que segmenten las publicaciones por periodo. Para hacer uso del plugin desde la sección destinada a las publicaciones del año 2025, por ejemplo, basta con incluir el siguiente fragmento directamente en la interfaz gráfica:

```
[imfd_publications year="2025"]
```

La Figura 8 muestra la implementación resultante.

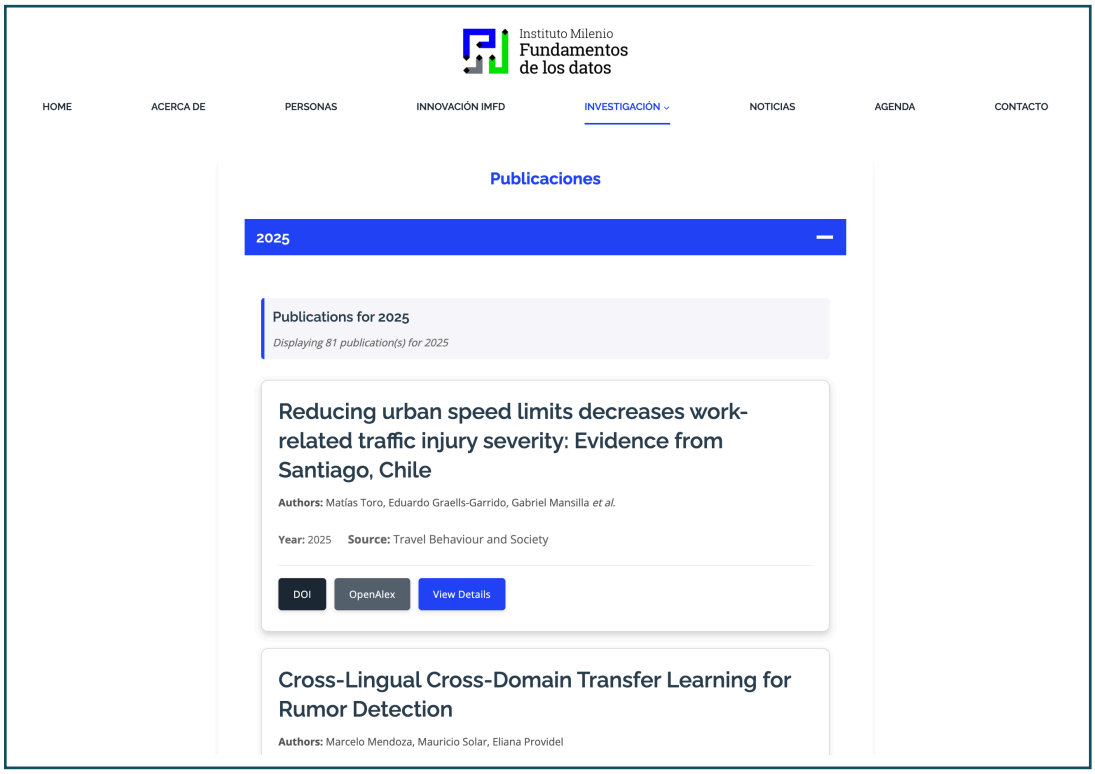

Figura 8: Listado de publicaciones para el año 2025 en el sitio web institucional.

Con la implementación de este plugin en WordPress se constituye el primer caso de uso funcional para evaluar la accesibilidad de la información científica del IMFD, estableciendo una conexión directa entre el grafo de conocimiento que centraliza la información y la plataforma web institucional para la visualización de los datos.

Desde la perspectiva del rendimiento, el tiempo de carga inicial presenta una limitación considerable, oscilando entre 45 segundos y 1 minuto durante la primera consulta. Esta latencia, aunque significativa para la experiencia de usuario, se resuelve mediante las estrategias de caché nativas de WordPress, que permiten cargas subsecuentes prácticamente instantáneas y garantizan una interacción fluida en usos posteriores.

Cabe mencionar que todos estos cambios fueron realizados sobre una copia del sitio web del IMFD[10], para evitar cualquier inconveniente con el sitio web principal.

5.2. Memorias anuales

El IMFD ha mantenido históricamente un proceso de recolección manual de información bibliográfica para la generación de memorias anuales, lo que representa una fuente de información confiable que ha sido validada directamente por los investigadores participantes en cada publicación. Esta práctica institucional consolidada proporciona un *ground truth*

[10]https://tset.imfd.cl/publicaciones/

ideal para evaluar la completitud y precisión del sistema automatizado implementado. Los documentos de memorias anuales siguen un formato estandarizado de planillas Excel con secciones dedicadas específicamente a artículos científicos, lo que permite realizar comparaciones cuantitativas entre la información históricamente recopilada y los datos consolidados automáticamente en Wikibase.

La generación automatizada de reportes de publicaciones cubre el segundo caso de uso planteado, implementándose mediante un DAG en Airflow que replica el formato tradicional de las memorias anuales. Este proceso, ilustrado en la Figura 9, consta de tres etapas principales: extracción de publicaciones por año específico desde Wikibase mediante una consulta SPARQL (como la que se muestra en el Código 6), generación de un archivo Excel con formato compatible con las memorias históricas, y almacenamiento en la infraestructura cloud institucional. La configuración del DAG incluye el año académico como variable en la interfaz de Airflow, permitiendo generar reportes para cualquier periodo simplemente modificando este parámetro y ejecutando el flujo con un clic. El proceso completo no toma más de 15 segundos, facilitando la comparación de estos reportes generados desde Wikibase con las memorias históricas del IMFD para evaluar la completitud de la información almacenada en el grafo de conocimiento.

Figura 9: DAG para generar reportes automáticamente.

Para realizar la evaluación se seleccionaron las memorias anuales correspondientes a los años 2021 y 2022, entregadas por el equipo IMFD a ANID, su agencia de financiamiento. A través de Airflow se generaron los reportes para esos mismos años. Se planteó medir la completitud mediante dos enfoques complementarios: primero, la proporción de DOIs de las memorias pasadas que fueron recuperados por el sistema automatizado, proporcionando una medida de la capacidad del sistema para replicar los resultados del proceso manual existente; segundo, el coeficiente de Jaccard para evaluar la similitud global entre ambos conjuntos, considerando tanto las coincidencias como las diferencias en cada sistema. El coeficiente de Jaccard mide la similitud entre dos conjuntos como la proporción de elementos compartidos respecto al total de elementos únicos en ambos conjuntos:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

donde A representa el conjunto de DOIs en las memorias históricas y B el conjunto de DOIs recuperados automáticamente desde Wikibase. Este coeficiente puede estar entre 0 y 1, donde valores cercanos a 1 indican alta similitud entre conjuntos y valores cercanos a 0 indican conjuntos disjuntos.

Cabe destacar que las memorias pasadas fueron parte de los datos que se incluyeron en la plataforma de Wikibase en sus inicios, por lo que al exportar información del grafo de conocimiento simplemente se estaría extrayendo la información de estas memorias sin considerar la influencia del sistema automatizado de extracción e integración. Ante esto, se exige en la consulta dirigida a la instancia de Wikibase que contenga un OpenAlex ID válido, de esta forma se asegura que el sistema fue capaz de extraer información sobre la publicación y la incluyó en el grafo.

Los resultados de este análisis se presentan en la Tabla 6. En ella se aprecia una tendencia por parte del sistema a recolectar información sobre un mayor número de publicaciones de las que consideran las memorias pasadas, aunque se sigue dejando fuera un número considerable de artículos que sí están presentes en las memorias anuales. Para el año 2021 el porcentaje de publicaciones que fue capaz de encontrar el sistema fue cercano el 70%, mientras que para el 2022 fue cercano al 60%. En cuanto al coeficiente de Jaccard, se observan valores de 0,35 y 0,31 para el 2021 y 2022, respectivamente. Dichos valores indican una intersección moderada entre ambos conjuntos de datos.

Año	**DOIs Memoria**	**DOIs Wikibase**	**Coincidencias**	**Porcentaje**	$\boldsymbol{J(A,B)}$
2021	137	242	99	72,26%	0,35
2022	132	199	78	59,09%	0,31

Tabla 6: Comparación de publicaciones entre memorias pasadas y Wikibase.

En un principio, dado el número de coincidencias que existen entre las publicaciones de ambos reportes, se esperaba un coeficiente de Jaccard mayor al obtenido, pero esto no es así dada la alta cantidad de publicaciones extra que considera el sistema automatizado. En la búsqueda de entender el origen de esta diferencia es que se caracterizaron los artículos presentes en los reportes según la fuente y tipo de publicación, como se presenta en la Tabla 7.

Año	**Total DOIs**		**ArXiv Papers**		**WoS/Scopus**	
	Memoria	**Wikibase**	**Memoria**	**Wikibase**	**Memoria**	**Wikibase**
2021	137	242	1	35	109	138
2022	132	199	0	33	90	96

Tabla 7: Segmentación de DOIs por fuente y tipo de publicación.

La segmentación por fuente y tipo de publicación revela diferencias significativas en los criterios de inclusión. En el caso de ArXiv, el sistema automatizado presenta una cobertura considerablemente más amplia, mientras las memorias históricas registraron apenas 1 publicación de ArXiv en 2021 y ninguna en 2022, Wikibase identificó 35 y 33 respectivamente, demostrando una mayor capacidad para capturar este tipo de contribuciones académicas.

No obstante, aunque ArXiv permite una rápida difusión del conocimiento académico, su naturaleza como repositorio de preprints implica que los trabajos allí publicados no han pasado por un proceso formal de revisión por pares, lo que puede explicar su exclusión en memorias pasadas debido a estándares más conservadores de validación. De manera similar, el análisis de publicaciones indexadas en WoS y Scopus muestra patrones diferenciados de cobertura. El sistema automatizado no solo captura un mayor volumen total de publicaciones, sino que también presenta una distribución más diversa en términos de indexación, sugiriendo criterios de inclusión más amplios que los empleados tradicionalmente en las memorias del Instituto. Es importante considerar que el flujo automatizado asume que todas las publicaciones de un investigador son pertinentes para el IMFD, mientras que en las definiciones de memorias pasadas es posible que los investigadores hayan excluido artículos en los que el IMFD no aparece acreditado como afiliación institucional, lo que podría explicar parcialmente estas diferencias en la cobertura observada.

5.3. Prueba de usabilidad

La prueba de usabilidad fue diseñada para evaluar la capacidad de los usuarios finales de interactuar efectivamente con la interfaz gráfica de Wikibase para realizar correcciones sobre la información recolectada automáticamente. Para la preparación de la prueba, se realizó una selección de 4 a 5 publicaciones por participante a partir de los reportes generados mediante el DAG implementado en Airflow. Se introdujeron errores en campos específicos como nombres de autores, títulos de artículos y fechas de publicación, replicando estos errores tanto en las planillas como en el grafo de conocimiento. Esta estrategia permitió crear un ambiente de prueba que simula errores reales que podrían requerir de intervención manual.

La evaluación se realizó con 8 participantes, incluyendo tanto investigadores del Instituto como miembros del equipo directivo del IMFD, representando así a los usuarios finales del sistema. Para los investigadores participantes, se tuvo especial cuidado de incluir sus propias publicaciones en las planillas asignadas, en un intento por aumentar el realismo de la evaluación y la motivación para realizar correcciones precisas. El flujo de evaluación consistió en tres etapas secuenciales: primero, los participantes debían revisar la planilla asignada para identificar errores en la información bibliográfica; segundo, utilizando enlaces directos provistos, navegar hasta la publicación correspondiente en la interfaz de Wikibase; y tercero, realizar las correcciones necesarias utilizando las herramientas de edición disponibles en la plataforma.

Una vez completado el proceso de corrección, los participantes respondieron una encuesta de usabilidad basada en el System Usability Scale (SUS) [10], un instrumento estándar para la evaluación de usabilidad de sistemas. La encuesta consta de 10 afirmaciones con escalas de respuesta de 5 puntos, desde «Totalmente en desacuerdo» hasta «Totalmente de acuerdo». Cabe destacar que en SUS hay preguntas positivas y negativas;

se buscan puntajes altos en preguntas impares y puntajes bajos en preguntas pares. El puntaje promedio obtenido por pregunta se encuentra en la Tabla 8. El resultado de la encuesta fue de 81 puntos sobre 100, lo que según los estándares de interpretación del SUS corresponde a un buen resultado. Este puntaje sugiere que el sistema de corrección implementado resulta intuitivo y manejable para los usuarios, permitiéndoles completar las tareas asignadas sin mayor esfuerzo.

Número	**Pregunta**	**Puntos**
1	Creo que me gustaría utilizar este sistema con frecuencia.	4,375
2	Encontré el sistema innecesariamente complejo.	1,75
3	Pensé que el sistema era fácil de usar.	4
4	Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.	2
5	Encontré que las diversas funciones de este sistema estaban bien integradas.	4,625
6	Pensé que había demasiada inconsistencia en este sistema.	1,375
7	Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.	4,25
8	Encontré el sistema muy complicado de usar.	1,375
9	Me sentí muy seguro usando el sistema.	4,125
10	Necesitaba aprender muchas cosas antes de empezar con este sistema.	2,25

Tabla 8: Puntajes obtenidos en la encuesta SUS.

El análisis detallado de las respuestas individuales del SUS revela algunos aspectos sobre la percepción de usabilidad del sistema. Las preguntas con mayor puntuación fueron la pregunta 5 y la pregunta 1, indicando que los usuarios perciben un sistema íntegro y deseable para uso continuo. Por el contrario, las puntuaciones más bajas correspondieron a las preguntas 6 y 8, que al ser preguntas formuladas negativamente, confirman la percepción positiva del sistema. Sin embargo, la pregunta 4 y la pregunta 10 sugieren que existe cierto nivel de dependencia inicial del soporte técnico y que los usuarios necesitan un periodo de familiarización con el sistema. Estos resultados son consistentes con un sistema que, si bien requiere una fase inicial de adaptación, logra establecer una experiencia de usuario satisfactoria.

Para complementar los datos cuantitativos, la encuesta incorporó componentes cualitativos con preguntas abiertas para explorar aspectos específicos de la interacción con el sistema, incluyendo preferencias sobre el mismo, elementos menos satisfactorios de la interfaz, sugerencias para funcionalidades adicionales, y comentarios generales sobre la plataforma. En base a estos comentarios es que se pudo confirmar una recepción general-

mente positiva del sistema, con usuarios destacando consistentemente su intuitividad. Sin embargo, se tienen problemas específicos de usabilidad que impactan la experiencia del usuario. Un desafío recurrente se relaciona con la edición de nombres de autores, donde los usuarios inicialmente intentan realizar modificaciones directamente desde la página de la publicación, sin comprender que deben navegar a la entidad específica del autor para realizar cambios. Esta confusión se ve agravada por la ausencia de feedback cuando las acciones de edición son bloqueadas, obligando a los usuarios a descubrir el flujo correcto mediante ensayo y error. Adicionalmente, se identificó un problema de propagación de cambios, donde las modificaciones realizadas en entidades de autores no se reflejan inmediatamente en las páginas de publicaciones relacionadas, incluso después de actualizar manualmente el navegador, lo que genera dudas sobre la efectividad de las correcciones realizadas. Estos hallazgos confirman que, aunque el sistema alcanza sus objetivos funcionales básicos, existe potencial significativo para mejorar la usabilidad mediante una mejor comunicación por parte del sistema.

Capítulo 6

Discusión y conclusión

Este capítulo presenta una reflexión crítica sobre el sistema desarrollado, analizando las decisiones de diseño, desafíos técnicos enfrentados y limitaciones identificadas durante la implementación. Se examina particularmente el balance entre procesos automatizados e intervención manual, aspecto clave en un sistema de gestión de conocimiento académico. La discusión fundamenta tanto las conclusiones del trabajo realizado como las direcciones futuras para el desarrollo continuo del sistema.

6.1. Desafíos y limitaciones

6.1.1. Cobertura de la información

El proceso de extracción de datos utiliza el DOI como identificador principal para enriquecer la información de OpenAlex con fuentes secundarias, sin embargo, un número considerable de publicaciones carece de este identificador. En una ejecución inicial del flujo se identificaron aproximadamente 3.500 publicaciones con DOI y poco más de 1.500 sin él. Para estas últimas, la información se almacena únicamente en el bucket de S3, ya que la ausencia de DOI impide tanto el enriquecimiento con fuentes externas como la integración con el grafo de conocimiento, al no existir un mecanismo confiable para determinar si la publicación ya se encuentra en Wikibase.

Existen limitaciones adicionales respecto a campos específicos de interés para el IMFD que no fue posible extraer automáticamente. El cuartil de revistas científicas no pudo incorporarse debido a las restricciones de acceso que impone Web of Science sin una licencia de pago, lo que es consistente con su modelo de negocio. De manera similar, el IMFD maneja un sistema de cuartiles interno para asociar cada publicación con sus proyectos emblemáticos, información que por su naturaleza no está disponible en fuentes externas.

Una limitación importante del modelado actual radica en que no se distingue entre tipos de publicaciones (journals, libros, artículos de conferencia), lo que imposibilita un análisis detallado sobre las actividades del Instituto y los lugares de publicación preferidos por sus investigadores. Aunque el sistema se encarga de generar entidades específicas para las revistas científicas, no se tiene el mismo cuidado con las conferencias. Si bien DBLP contiene información valiosa sobre series de conferencias y journals, actualmente se extrae solo un atributo general que no permite diferenciar por tipo de publicación. Además, según la documentación oficial del esquema RDF de DBLP[11], las propiedades específicas para conferencias y journals serán eventualmente eliminadas una vez que se modelen apropiadamente. Esta limitación se ve agravada por el hecho de que DBLP cubre únicamente el área de ciencias de la computación.

6.1.2. Integración de los datos

Una de las decisiones de diseño más críticas del sistema fue la elección de la estrategia `APPEND_OR_REPLACE` sobre `KEEP` para el manejo de claims en Wikibase. Esta decisión, aunque vela por la completitud de los datos, presenta trade-offs importantes. Con la estrategia `KEEP`, si una publicación inicialmente tenía dos autores registrados y posteriormente se descubre un tercer autor, esta información adicional no se incorporaría automáticamente. Sin embargo, `APPEND_OR_REPLACE` puede agregar más información para papers ya corregidos manualmente por investigadores, creando un conflicto entre la automatización y la validación humana. Finalmente se optó por un enfoque más flexible con la información que entra al grafo, antes que desechar la información que se extrae de las fuentes externas.

Los datos legados presentes en la plataforma antes de la implementación del sistema automatizado presentan desafíos particulares, especialmente en la gestión de autores donde pueden existir entidades duplicadas que se refieren a la misma persona. Para evitar la duplicación de entidades en Wikibase, es que se realiza una consulta para obtener los OpenAlex IDs ya existentes en el grafo de conocimiento, pero existen entidades que no contaban con este identificador y que van a ser creadas nuevamente en el grafo. El sistema actual no implementa una resolución retroactiva de duplicados, aunque es autosuficiente para evitar duplicación en datos nuevos.

El grafo de conocimiento corresponde a una entidad separada del servicio de consultas SPARQL. Por lo mismo, es que se cuenta con un mecanismo para sincronizar la información que no es inmediato. Se realizaron pruebas que indicaron que el servicio de consultas detecta cambios en Wikibase con un retraso de 5 a 10 segundos posteriores a la inserción de nuevos statements. La carga de información en el grafo por primera vez toma cerca de 2 horas para procesar alrededor de 3500 publicaciones, con un promedio de 2 segundos por publicación, tiempo que varía según la cantidad de información y relaciones asociadas. El problema con esto es que el DAG en ocasiones puede ejecutarse más rápido que la

[11] https://dblp.org/rdf/docu/

actualización del servicio de consultas, creando entidades duplicadas cuando publicaciones consecutivas comparten información común, como autores o instituciones.

Un desafío técnico surgió al procesar los abstracts de las publicaciones. Wikibase por defecto limita los elementos de tipo texto a 400 caracteres, impidiendo poder incluir cualquier abstract que superara esta extensión. Para poder solucionar este problema fue necesario cambiar la configuración por defecto a 2 mil caracteres directamente en el contenedor de Docker que contiene al servicio. Ante cualquier caso borde, se trunca el abstract extraído de OpenAlex para que no supere este largo.

6.2. Trabajo futuro

Estrechamente relacionado con el desafío impuesto por los datos legados, es que a futuro se propone un mecanismo de detección y resolución retroactivo de duplicados, permitiendo abordar inconsistencias entre los datos mediante consultas SPARQL que identifiquen entidades potencialmente duplicadas, junto al método `merge_items` que provee WikibaseIntegrator para unificar dos entidades en una sola.

La implementación de una funcionalidad de «lista negra» permitiría excluir del proceso de extracción y carga a publicaciones que han sido corregidas manualmente por investigadores o que no son relevantes para el contexto institucional. Esta capacidad es particularmente importante para investigadores recién incorporados al IMFD que poseen un historial de publicaciones previo a su afiliación institucional, así como para filtrar preprints de ArXiv o cualquier material que no cumpla con los criterios de inclusión establecidos por el Instituto.

Para abordar la clasificación de publicaciones según los proyectos emblemáticos del IMFD, se plantean tres enfoques complementarios. El primero consiste en entrenar algoritmos de machine learning supervisado utilizando las memorias anuales que ya contienen etiquetas de proyecto, clasificando nuevas publicaciones basándose en sus abstracts. Una segunda alternativa aprovecharía los tópicos que provee OpenAlex para cada publicación, información que actualmente no es utilizada por el sistema. Finalmente, se podría implementar la clasificación mediante modelos de lenguaje grande (LLM) que analicen tanto los abstracts como los tópicos de OpenAlex para determinar la correspondencia con proyectos emblemáticos.

6.3. Conclusión

El sistema desarrollado representa un avance significativo en la gestión automatizada de información bibliográfica para el IMFD, cumpliendo exitosamente con el objetivo general de desarrollar e implementar un sistema que centraliza, actualiza y optimiza los flujos de información institucional. La solución logró integrar efectivamente múltiples fuentes de datos

bibliográficos (OpenAlex, DBLP, Web of Science y Scopus) en un grafo de conocimiento estructurado basado en Wikibase, estableciendo un repositorio centralizado que automatiza completamente el proceso de recolección y estructuración de datos. Esta implementación reduce significativamente el esfuerzo manual requerido para la generación de reportes con información de las publicaciones de investigadores asociados al Instituto, mientras que el procesamiento incremental optimiza la eficiencia operacional y la integración con el sitio web institucional mejora sustancialmente la accesibilidad pública de la información. La arquitectura modular del sistema orquestado en Airflow facilita su mantenimiento y extensión, proporcionando una base sólida para desarrollos futuros, mientras que el uso de estándares abiertos como SPARQL y Wikibase asegura interoperabilidad y sostenibilidad a largo plazo.

La evaluación del sistema, actualmente desplegado en producción, confirma su efectividad técnica y usabilidad práctica. Las pruebas SUS demostraron buenos niveles de usabilidad, mientras que el análisis comparativo mostró que el sistema identifica aproximadamente 70% y 60% de las publicaciones reportadas manualmente para el 2021 y 2022, respectivamente, pero captura un volumen considerablemente mayor de publicaciones adicionales, evidenciando diferencias en los criterios de inclusión entre el sistema automatizado y el proceso manual de recolección del Instituto.

El desarrollo de este sistema ilustra la complejidad inherente en la gestión automatizada de datos bibliográficos, donde la precisión y completitud debe balancearse con la flexibilidad para incluir correcciones humanas y necesidades institucionales específicas. La experiencia ganada sugiere que el éxito de estos sistemas no depende solo de la solidez técnica sino también de la comprensión profunda de los procesos que se buscan optimizar. El sistema establecido proporciona una base robusta para la evolución continua de la gestión de conocimiento del IMFD, adentrando a la institución en el uso de tecnologías de datos semánticas para representar y modelar su información interna.

Bibliografía

[1] O. Lassila y R. R. Swick, «Resource Description Framework (RDF) Model and Syntax Specification», 1999, [En línea]. Disponible en: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[2] D. Brickley y R. Guha, «RDF Schema 1.1», feb. 2014. [En línea]. Disponible en: https://www.w3.org/TR/rdf-schema/

[3] A. Hogan *et al.*, «Knowledge Graphs», *ACM Comput. Surv.*, vol. 54, n.º 4, jul. 2021, doi: 10.1145/3447772.

[4] D. Vrandečić y M. Krötzsch, «Wikidata: a free collaborative knowledgebase», *Commun. ACM*, vol. 57, n.º 10, pp. 78-85, sep. 2014, doi: 10.1145/2629489.

[5] D. Diefenbach, M. D. Wilde y S. Alipio, «Wikibase as an Infrastructure for Knowledge Graphs: The EU Knowledge Graph», en *The Semantic Web – ISWC 2021*, 2021, pp. 631-647.

[6] J. Priem, H. Piwowar, y R. Orr, «OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts», 2022, [En línea]. Disponible en: https://arxiv.org/abs/2205.01833

[7] M. Ley, «DBLP: some lessons learned», *Proc. VLDB Endow.*, vol. 2, n.º 2, pp. 1493-1500, ago. 2009, doi: 10.14778/1687553.1687577.

[8] D. Dessí, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, y E. Motta, «CS-KG: A Large-Scale Knowledge Graph of Research Entities and Claims in Computer Science», en *The Semantic Web – ISWC 2022*, U. Sattler, A. Hogan, M. Keet, V. Presutti, J. P. A. Almeida, H. Takeda, P. Monnin, G. Pirrò, y C. d'Amato, Eds., Cham: Springer International Publishing, 2022, pp. 678-696.

[9] M. R. Ackermann, H. Bast, B. M. Beckermann, J. Kalmbach, P. Neises, y S. Ollinger, «The dblp Knowledge Graph and SPARQL Endpoint», *TGDK*, vol. 2, n.º 2, pp. 1-23, 2024, doi: 10.4230/TGDK.2.2.3.

[10] J. Brooke, «SUS: A quick and dirty usability scale», *Usability Eval. Ind.*, vol. 189, 1995.