UNIVERSIDAD
DE MÁLAGA

TESIS DOCTORAL
Tecnologías Informáticas

Explotación de la semántica de dominio orientada a la integración, estandarización y análisis de datos

E.T.S.I. Informática
R.D. 99/2011

Autor

Manuel Paneque Romero

Directores

Dr. María del Mar Roldán García
Departamento
Lenguajes y Ciencias de la Computación
Universidad de Málaga

Dr. José Manuel García Nieto
Departamento
Lenguajes y Ciencias de la Computación
Universidad de Málaga

2024

AUTOR: Manuel Paneque Romero

https://orcid.org/0000-0002-7973-3746

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña MANUEL PANEQUE ROMERO
Estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: EXPLOTACIÓN DE LA SEMÁNTICA DE DOMINIO ORIENTADA A LA INTEGRACIÓN Y ANÁLISIS DE DATOS

Realizada bajo la tutorización de ISMAEL NAVAS DELGADO y dirección de MARÍA DEL MAR ROLDÁN GARCÍA Y JOSÉ MANUEL GARCÍA NIETO (si tuviera varios directores deberá hacer constar el nombre de todos)

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 27 de DICIEMBRE de 2023

Fdo.: MANUEL PANEQUE ROMERO Doctorando/a	Fdo.: ISMAEL NAVAS DELGADO Tutor/a
Fdo.: MARÍA DEL MAR ROLDÁN GARCÍA Y JOSÉ MANUEL GARCÍA NIETO	

DECLARACIÓN DE DIRECCIÓN Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

Dr. D. José Manuel García Nieto y Dra. Dña. María del Mar Roldán García, profesores doctores del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, certifican que el doctorando Manuel Paneque Romero, ha realizado bajo su dirección y tutorización en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, el trabajo de investigación correspondiente a su TESIS DOCTORAL titulada:

Explotación de la semántica de dominio orientada a la integración, estandarización y análisis de datos

En dicho trabajo se han hecho aportaciones originales que han dado lugar a las publicaciones en coautoría en revistas y comunicaciones a congresos que avalan la tesis, las que no han sido utilizadas en tesis anteriores.

En Málaga, a 27 de DICIEMBRE de 2023

Fdo.: JOSÉ MANUEL GARCÍA NIETO Director de tesis	MARÍA DEL MAR ROLDÁN GARCÍA Directora de tesis

DECLARACIÓN DE TUTORIZACIÓN Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D. ISMAEL NAVAS DELGADO, profesor doctor del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga

DECLARA QUE:

D. MANUEL PANEQUE ROMERO, estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS ha realizado bajo su tutorización la tesis presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: EXPLOTACIÓN DE LA SEMÁNTICA DE DOMINIO ORIENTADA A LA INTEGRACIÓN, ESTANDARIZACIÓN Y ANÁLISIS DE DATOS.

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo. Así mismo, las publicaciones en coautoría que avalan dicha tesis no forman parte de otra tesis doctoral en la Universidad de Málaga ni en ninguna otra universidad.

En Málaga, a 27 de DICIEMBRE de 2023

Fdo.: ISMAEL NAVAS DELGADO
Tutor de tesis

Edificio Pabellón de Gobierno. Campus El Ejido. 29071
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10
E-mail: doctorado@uma.es

Índice general

Agradecimientos

La presente Tesis Doctoral se realizó bajo la tutorización de Ismael Navas Delgado, junto con la codirección de María del Mar Roldán García y José Manuel García Nieto, a los que le quiero agradecer su paciencia, apoyo y tiempo para que aflorara en mí el conocimiento que se presenta en este manuscrito.

A los compañeros del grupo Khaos, gracias por tantas horas que compartimos. Quiero hacer mención especial a Cristóbal Barba González, Sandro Hurtado Requena y Antonio Benítez Hidalgo por su apoyo en diferentes etapas de este camino.

Aagradecer también a las personas que he conocido en eventos académicos y de transferencia de conocimiento, nacionales e internacionales; por ayudar a ampliar la visión, realizar trabajos conjuntos y estar al día del estado del arte.

Agradecer a todos los familiares, amigos y conocidos su apoyo y empatía en los momentos alegres y difíciles.

Gracias al Gobierno de España, la Unión Europea, la Junta de Andalucía y la Universidad de Málaga por impulsar y apoyar la investigación y transferencia del conocimiento.

Quiero terminar agradeciendo esta Tesis en memoria de Miguel Paneque. Gracias por desde mi infancia y junto con mis padres, despertar en mí la curiosidad por la Ingeniería en general y posteriormente en la rama específica de la Ingeniería Informática.

Abstract

Introduction

Big Data has become an inexhaustible source of information thanks to the increasing amount of data generated every day, driven by the rise of sensor networks and mobile devices. Proper management and analysis of these data can provide more objective, fact-based insights for decision-making, rather than relying solely on intuition.

Big Data analytics involves examining large datasets with the aim of uncovering patterns, trends and correlations that may not be obvious to the naked eye. The use of Machine Learning (ML) techniques, makes it possible to discover patterns and learn models from the data. This can be especially useful for identifying patterns in data and predicting future trends.

In addition, one of the biggest challenges in data analytics today is the use of domain and context knowledge in the process of analysis. Domain knowledge refers to the general knowledge of a field or environment in which data analysis methods are applied. The inclusion of this contextual knowledge can significantly improve the analysis process, allowing the selection of appropriate information, features or techniques, the reduction of the search space and the representation of the output in a more understandable way.

In a specific domain, the sources of information can vary significantly, including the possibility of obtaining information through consultation with experts or the automatic extraction of data from different sources, such as data repositories, documents in several formats (XML, JSON, CSV, etc.) or websites. Although there are techniques available, such as ontology learning [1] and schema inference [2], to carry out this extraction, the lack of a standard in the publication of the data, as well as the knowledge associated with such data, entails an effort to integrate it and make them accessible and interoperable.

In this sense, the Semantic Web is a technological concept that emerged in the early 2000s with the vision of transforming the web into a more intelligent and meaningful, data-oriented platform, where machines could understand content in a similar way just like humans do. Using these techniques, semantically annotated distributed knowledge graphs are generated, which constitute a key tool for the integration, management and analysis of large amounts of data in the era of Big Data. These graphs provide a structure for organizing and connecting data in a semantic network, allowing for greater understanding and contextualization of the data [3]. By including this contextual information, the quality and accuracy of data analysis can be significantly improved, which is essential for data-driven decision making in a variety of sectors and disciplines.

Although significant progress has been made since then, several challenges still exist currently in the field of the Semantic Web. Some of the most important are: data representation and standardization, scale and performance, data integration, data quality, privacy and security, interoperability, update and evolution of ontologies.

Due to these challenges, Semantic Web technologies remain an area of active research and development, which is gaining momentum in recent years thanks to the recent generation of European

Data Spaces[1], in which they can make significant contributions by providing a framework for the organization, exchange and understanding of data more effectively. Among some of the possible contributions of the Semantic Web to data spaces, it is worth mentioning, for example: data enrichment, access and discovery of data, open data management, data governance, citizen science, multi-language and regulation issues.

Motivation

The role of the Semantic Web has become increasingly relevant due to its ability to organize and relate information more effectively, which is why it motivates research into new methods and real applications based on this technology. Among these applications we can mention, from internet search, to artificial intelligence and business decision making.

In this sense, it is worth highlighting the use of semantic knowledge graphs to represent information in a structure of nodes and relationships, which facilitates the organization of complex data and its connection in a coherent network. This is particularly useful in managing structured and unstructured data.

Likewise, this technology has great potential to support data analysis processes based on Artificial Intelligence (AI) methods, since they help model and represent knowledge in multiple applications, from computer vision and autonomous decision making, to the interpretation of algorithmic outputs. In particular, within Natural Language Processing (NLP), semantic graphs are essential, as they help to understand the semantics of words and sentences, with important contributions in applications such as chatbots, virtual assistants and machine translation. Furthermore, continuing along this line, it is important to explore the possible relationship between chatbots based on large linguistic models (LLM), such as ChatGPT [2], and the concept of the Semantic Web. This symbiosis can occur for several reasons: first, LLMs provide a foundation for powerful natural language processing based on their understanding of the syntax and semantics of sentences; for example, they understand the underlying semantics of multiple variations of the same phrase; Second, the Semantic Web began as a variation of the web explicitly constructed from sentences that use hyperlinks to express the syntax and semantics of the sentence in a machine-readable way; The effect is an unlimited set of structured data that manifests a global graph of entity relationships (rather than a network), comprising machine-readable entity relationship-type semantics.

From an orthogonal viewpoint, the use of these technologies is already a reality in the automation of business and scientific processes based on data, by serving as a common thread to model workflows, resources and dependencies. This improves efficiency, supports data traceability, and reduces errors or inconsistencies when connecting processing and analysis components. The TITAN [4] platform represents progress in this sense, as it is already being used in citizen science projects at European level.

All these reasons serve as inspiration and motivation for this PhD Thesis, highlighting the need to go one step further in the generation of new ontologies and semantic models. The idea is to extract common experiences and conclusions when exploring their potential in real-world applications in heterogeneous knowledge domains, as well as their ability to represent knowledge in a structured way and understand the relationships between concepts, in a way that supports development processes of artificial intelligence in data analysis. Based on this, the following research questions are formulated, which have guided us in the process of modeling, experimentation and validation of scientific proposals in a set of use cases.

[1] European Data Spaces website (last visited 19-10-2023)`https://dataspaces.info/common-european-data-spaces`

[2] ChatGPT Website`https://chat.openai.com/`

Research questions

- **Q1**: How can semantic models and knowledge graphs be leveraged to improve data integration, harmonisation and standardisation?
- **Q2**: How can semantic models and knowledge graphs be combined with other emerging technologies, such as artificial intelligence and machine learning, to improve the quality and value of data analysis in e-learning environments?
- **Q3**: How can semantic inference techniques automate data reconciliation and financial solvency determination processes in data-driven open banking services?
- **Q4**: How can semantic models and knowledge graphs be used to improve security and privacy in NoSQL databases?

Objectives and phases

This PhD Thesis focuses on the integration of different sources of information and the improvement of data analysis for decision making through the use of emerging Semantic Web techniques. To achieve this general objective, a common framework is proposed through the generation of models and ontologies that facilitate the semantic integration of data and its representation in the form of a graph, which facilitates data analysis and knowledge extraction.

Specifically, a series of specific objectives are highlighted below, aligned with the main objective, although focusing on the selected application domains:

- **Objective 1: Design and implement ontologies and knowledge graphs in specific domains,** which allow the implementation of data integration, harmonisation and standardisation solutions, as well as the exploitation of their reasoning and data analysis capacities in artificial intelligence processes.
- **Objective 2: Following a semantic model, feed through a knowledge graph a set of Machine Learning algorithms capable of analysing implicit interaction patterns in e-Learning Management Systems (LMSs).** The generated knowledge graph will consider the integration of several real-world and academic data sources about the interaction of students and teachers in university communities.
- **Objective 3: Define a knowledge graph that aligns PSD2 open banking transactions with invoice information to perform bank solvency studies.** Furthermore, it is intended to obtain a semantic rule system to show how the financial solvency classification of client entities and transaction concept suggestions can be inferred from the proposed semantic model.
- **Objective 4: To define an ontology-driven framework for designing secure graph-based databases, which facilitates the rapid migration of security rules by deriving specific measures for each underlying technology.** The aim is to provide database designers with methods to check through ontological reasoning whether security rules are consistent.

In general, the phases to achieve these objectives include: the definition of domain information requirements, the selection of relevant data sources, the design and development of semantic models and ontologies, the adaptation of advanced data analysis techniques and the validation of the results

through specific use cases. The latter is important to demonstrate the usefulness and effectiveness of the proposed models and frameworks in actual situations.

In particular, the phases of the common methodology followed in this Thesis are presented, giving homogeneity to the conception of each design case, in the specific application domains:

- **Phase 1: Identification of relevant sources of information,** considering other existing related ontologies, such as data sets and specification of requirements.

- **Phase 2: Design of the semantic model.** This phase involves the definition of the concepts and relationships that will be used to represent the information and the structure of these elements in an ontology.

- **Phase 2: Implementation of the semantic model.** In this phase, the semantic model is implemented through the use of specific tools for the creation of ontologies. Linking with existing related ontologies and modification, where appropriate, to adapt to new requirements is contemplated.

- **Phase 3: Populating the knowledge graph and persistence methods.** In this phase, we proceed to integrate the different sources of information identified in the first phase. This may involve performing extraction, transformation and loading tasks, data mappings and integration, to adapt them to the defined semantic model. The database is also established to store the graph and EndPoint services for querying.

- **Phase 4: Validation of the semantic model.** The semantic model developed is validated through specific use cases. This phase may include evaluating the quality of the integrated information, through pre-designed queries, verifying the consistency of the model, and evaluating the performance of tools and applications that use the semantic model. After this evaluation, it could be considered to return to Phase 2 to reimplement and refine the model.

- **Phase 5: Selection and application of advanced data analysis techniques.** This phase involves the selection and application of the most appropriate advanced data analysis techniques for the study domain and the integrated data. This involves exploring the data to identify patterns and relationships, generating specific queries for data nurturing, selecting and adjusting prediction models, and evaluating the accuracy and robustness of the models.

- **Phase 6: Making informed decisions.** In this phase, the information obtained from the integration of information sources and advanced data analysis is used to make informed decisions in the study domain. This may involve identifying relevant patterns and trends, evaluating different scenarios and options, along with selecting the best solution based on the objectives and constraints of the problem at hand.

- **Phase 7: Evaluation of results and continuous improvement.** Finally, in this phase the results obtained from the integration of information sources and the use of advanced data analysis techniques are evaluated. This could lead to identify areas of improvement and learning opportunities to leverage decision making and prediction accuracy in future projects or similar situations.

Scientific contributions

As a result of the work carried out in this PhD Thesis, a series of ontologies have been generated, namely: e-LION [3], OBO [4] and OntoSecurityGraphDB [5]; as well as 3 impact scientific publications, which include the specific contributions detailed in Chapters 3, 4 and 5. These publications have been evaluated and accepted in international scientific journals indexed in the Journal of Citation Report (JCR), two of them being in quartile Q1 (categories of Computer Science) and another one in Q2 (category of Multidisciplinary Engineering). In addition, presentations have been made at the Software and Database Engineering Conference (JISBD), in its 2022 and 2023 editions.

Scientific publications

1. Manuel Paneque, María del Mar Roldán-García, José García-Nieto, ***e-LION: Data Integration Semantic Model to Enhance Predictive Analytics in e-Learning***, *Expert Systems With Applications* (Q1, *Computer Science, Artificial Intelligence*, Position: 21/145, IF: 8.665, DOI: `https://doi.org/10.1016/j.eswa.2022.118892`).

   In this study, an innovative approach to predictive analytics in e-learning platforms is proposed. It uses an ontology-based semantic data integration model, called e-LION, to integrate and analyze data from several sources in online learning environments. This semantic approach is novel as it allows for a deeper and more accurate understanding of the data, leading to better predictions about student performance and motivation. The implementation of the e-LION model aims to improve the effectiveness of e-learning through a better understanding of the factors that affect student behaviour.

2. Manuel Paneque, María del Mar Roldán-García, José García-Nieto, ***A Semantic Model for Enhancing Data-Driven Open Banking Services***, *Applied Sciences* (Q2, *Engineering, Multidisciplinary*, Rank: 39/92, Impact Factor: 2.838, DOI: `https://doi.org/10.3390/app13031447`).

   This work focuses on improving data-driven open banking services. It proposes a semantic model to integrate and analyze data from different sources in the open banking environment. This model enables a deeper and more accurate understanding of data, leading to better business decisions and better services for customers. The implementation of the semantic model aims to improve the efficiency and effectiveness of open banking services.

3. Manuel Paneque, María del Mar Roldán-García, Carlos Blanco, Alejandro Maté, David G. Rosado, Juan Trujillo, ***An Ontology-based Secure Design Framework for Graph-based Databases.*** *Computer Standards & Interfaces*, (Q1, in *Computer Science, Sotware Engineering*, Rank: 24/110, Impact Factor: 3.721, DOI: `https://doi.org/10.1016/j.csi.2023.103801`.

   This work addresses the importance of security in graph-based databases. In today's digital world, databases are a critical resource for most organizations, and the security of this data is essential to ensure the privacy and confidentiality of information. However, graph-based databases present unique security challenges due to their highly connected structure and ability to represent complex relationships between entities. A novel ontology-based security design framework is therefore presented to address these challenges and provide an effective

[3] `https://ontologies.khaos.uma.es/e-lion`
[4] `https://ontologies.khaos.uma.es/obo`
[5] `https://proyectoaether.github.io/OntoSecurityGraphDB/index-en.html`

solution to the security design of graph-based databases and their consistency. This framework uses ontology concepts to model the database structure and define appropriate access rights and security control. In addition, the semantic model is also verified by modeling security in a database and automatically generating the code to implement the model and security in different graph-oriented database management systems.

Conferences

Two contributions to conference are made in the "Jornadas de Ingeniería del Software y Bases de Datos (JISBD)" in its 2022 and 2023 editions. These contributions are presented in the respective special sessions of "Relevant Article", in which the proposals are described scientific developments developed in the previous articles.

Conclusions

Throughout its chapters, this PhD Thesis addresses an innovative and applied semantic approach for the integration and analysis of data in different areas of knowledge. A series of specific conclusions are obtained for each application domain, which in turn lead us to draw general conclusions aimed at answering the research questions posed in the introduction.

The following are extracted as the main specific contributions:

- In Chapter 3, the proposed semantic approach is shown to adequately integrate data from *"e-Learning"* management systems, allowing advanced query and constituting a well-founded knowledge graph to improve information analysis in the context of LMSs. This leads the proposed e-LION ontology to provide added scientific value, which in the context of the current state of the art (as explained in Section 3.2), allows semantic connection with other related ontologies and vocabularies, thus promoting the generation of extensive linked data in the domain of *"online"* learning. Therefore, the proposal can be used at the core of a data consolidation strategy for future applications, where current LMSs and other academic data sources are systematically queried to support teachers with advanced analytics and visualizations of what is happening in their subjects. Likewise, it allows students' activities to be analyzed in the context of overall performance in a given course, which would provide them with a global perspective of their performance in the course, thus promoting their proactive learning.

- In the context of open banking (Chapter 4), the Open Bank Ontology (OBO) is introduced, which defines a semantic model for the consolidation and annotation of banking transaction data according to the paradigm defined by the PSD2 regulations. This model enables SPARQL queries and SWRL reasoning, facilitating data reconciliation and customer classification based on debt patterns. This ontology fosters extensions and federations with related knowledge graphs in the Fintech domain. It should be noted that the proposed ontology constitutes the first attempt to model banking movements and activities with special attention to the European PDS2 standard. Therefore, new extensions of this work are expected, including federation with other related knowledge graphs in the domain of Fintech applications and standards.

- Likewise, Chapter 5 addresses security in NoSQL databases with a high abstraction framework focused on secure design. This framework, which includes an ontology for modeling security concepts and policies, provides an essential methodology for designing secure NoSQL

databases, regardless of the specific technology. In addition, the ontology layer allows analysis and error detection in the design phase, optimizing secure implementation.

As general conclusions, throughout this PhD Thesis report we show how semantic models based on ontologies, together with knowledge graphs, are powerful and flexible tools to transform dispersed and heterogeneous data into homogeneous, structured and related knowledge. This supports the integration, harmonization and standardization of data, in turn answering the first research question (Q1) set out in the introduction.

However, the precise definition of "data mappings" from the original sources to a unified representation model is necessary, resulting in a knowledge graph formally defined by the specific domain ontology. This mapping provides the key to translate the diversity of formats, structures and semantics of the source data into a standardized format, thus facilitating its integration and subsequent analysis. In fact, the development of such "mappings" requires a deep understanding of the specific domain and the intrinsic characteristics of each data source. It is necessary to define rules and transformations that allow information to be converted from its original format to an understandable and coherent format in the knowledge graph. This mapping process not only involves the translation of schemas and data types, but also the appropriate semantic mapping, where concepts from the data sources are precisely related to concepts in the knowledge graph. Therefore, data reconciliation and its organization are automated for exploitation in different domains, such as those addressed in this work, e-Learning, open banking and security, thus answering research questions Q2, Q3 and Q4. respectively.

As an additional contribution, the proposed ontologies have been designed and implemented following the FAIR principles. Mapping functions of the original data have also been defined for each domain, as we propose in the research challenges.

In short, these conclusions reflect significant advances in the semantic integration of data and its application in diverse contexts, pointing out the usefulness and relevance of the proposed models. Additionally, they establish solid foundations for future research and extensions in each domain, contributing to continued progress in advanced data integration and analysis.

Future work

As future lines of research in general, we plan to continue with this proposal of integration of heterogeneous data to improve access and its connection with analysis and explainability techniques. The main objective is to use the semantic layer as support for the interpretation of analysis for the expert in the domain of knowledge, thus improving both the quality of the results and the transparency in the use of the analysis algorithms.

Another important aspect to consider in the near future is to design new ontologies and data strategies based on Semantic Web technologies, to support the generation of "Data Spaces". These spaces are being articulated as key pieces for the standardization and regulation of the new data economy, both in industrial and academic, as well as administrative and organizational environments. In this sense, initiatives such as "GAIA-X" are already being carried out[6] and "European Data Spaces"[7] , of great relevance at a national and European level, in which the semantic layer and data linking are considered a fundamental requirement in these spaces.

Following along this line, a symbiotic relationship between conversational bots based on large linguistic models (LLM), such as ChatGPT, and the concept of semantic web (public, private or hybrid)[1]. This symbiosis is due to the reasons stated above in ChatGPT's answer, in the way that LLMs provide a foundation for powerful natural language processing based on your understanding

[6]GAIA-X website (last visit 10-16-2023) `https://gaia-x.eu/`

[7]Common European Data Spaces website (last visited 10-16-2023) `https://dataspaces.info/common-european-data-spaces/`

of the syntax and semantics of sentences; for example, they understand the underlying semantics of multiple variations of the same phrase. A semantic web is simply a variation of the web explicitly constructed from phrases that use hyperlinks to express the syntax and semantics of the phrase in a machine-readable form; The net effect is an unlimited collective of structured data that manifests a global graph of entity relationships (rather than a network), comprising machine-readable entity relationship-type semantics.

Furthermore, considering different aspects of the areas of knowledge related to the contributions proposed in these studies, different lines of research have been identified for future work. This section presents the most notable ones:

- In the context of *"e-Learning"*, it is proposed to include more data from other learning management systems *"online"*, as well as to update the e-LION ontology to incorporate new attributes relevant from different perspectives of LMSs. In this sense, another future activity is the ontological alignment of many others not only in the domain of educational knowledge, but also in different domains, such as: social networks, health-related behaviors of Covid-19 users, demographic evolution and social.

- In the Fintech domain, it is planned to integrate more data from different invoice management systems and update the OBO ontology to incorporate new relevant attributes from different perspectives, such as: opinions on social networks, behavioral traits of the company in its commercial relationships with clients, etc. This new knowledge will allow new analyzes to be carried out, taking into account more factors and actors.

- Finally, within the field of security, it is intended to extend the proposal of this Thesis to all types of databases. Currently, the proposed methodology has been developed and validated using a graph-oriented database as a case study. However, it is recognized that there are different types of databases with diverse characteristics and structures. Therefore, it is intended to investigate and adapt the methodology for its application in a wide range of databases, including relational databases, NoSQL databases and distributed databases. This will involve studying the peculiarities and challenges associated with each type of database, and adapting the methodology to effectively address its particularities. By expanding the proposal to all types of databases, it is expected that this research will have a greater impact in the field of data management and provide more general solutions applicable to a wide variety of database environments and systems.

Parte I
Introducción y contexto

Capítulo 1

Introducción

El Big Data se ha convertido en una fuente inagotable de información gracias a la creciente cantidad de datos generados a diario, impulsados por el auge de las redes de sensores y los dispositivos móviles. La adecuada gestión y análisis de estos datos pueden proporcionar una visión más objetiva y basada en hechos para la toma de decisiones, en lugar de depender exclusivamente de la intuición.

El análisis de Big Data implica el examen de grandes conjuntos de datos con el objetivo de descubrir patrones, tendencias y correlaciones que pueden no ser evidentes a simple vista. La utilización de técnicas de aprendizaje automático, como el Machine Learning (ML), permite descubrir patrones y aprender modelos a partir de los datos. Esto puede ser especialmente útil para identificar patrones en los datos y prever tendencias futuras.

Además, uno de los mayores desafíos en el análisis de datos en la actualidad es la utilización del conocimiento del dominio y del contexto en el proceso de análisis. El conocimiento del dominio se refiere al conocimiento general de un campo o entorno en el que se aplican los métodos de análisis de datos. La inclusión de este conocimiento contextual puede mejorar significativamente el proceso de análisis [5], permitiendo la selección de la información, características o técnicas adecuadas, la reducción del espacio de búsqueda y la representación de la salida de una manera más comprensible.

En un dominio específico, las fuentes de conocimiento pueden variar significativamente, incluyendo la posibilidad de obtener información a través de la consulta a expertos o la extracción automática de datos de diversas fuentes, como repositorios de datos, documentos en varios formatos (XML, JSON, CSV, etc.) o sitios web. Aunque existen técnicas disponibles, como el ontology learning [1] y la inferencia de esquemas [2], para llevar a cabo esta extracción, la falta de un estándar en la publicación de los datos y el conocimiento asociado a dichos datos conlleva un esfuerzo para su integración y hacer que los datos sean accesibles e interoperables.

En este sentido, la Web Semántica es un concepto que surge a principios de la década de 2000 con la visión de transformar la web en una plataforma más inteligente y significativa, orientada a los datos, donde las máquinas pudieran comprender el contenido de una manera similar a como lo hacen los humanos. Mediante estas técnicas se generan grafos de conocimiento distribuidos anotados semánticamente, que constituyen una herramienta clave para la integración, gestión y análisis de grandes cantidades de datos en la era del Big Data. Estos grafos proporcionan una estructura para organizar y conectar datos en una red semántica, lo que permite una mayor comprensión y contextualización de los datos [3]. Al incluir esta información contextual, se puede mejorar significativamente la calidad y precisión del análisis de datos, lo que es esencial para la toma de decisiones basada en datos en una variedad de sectores y disciplinas.

Aunque se han logrado avances significativos desde entonces, todavía existen varios desafíos en la actualidad en el campo de la Web Semántica. Algunos de los más importantes son:

- Representación y estandarización de datos: comprendiendo la creación y adopción de estándares para la representación de datos de manera semántica. Lenguajes como RDF (Resource Description Framework) y OWL (Web Ontology Language), que decribiremos más en detalle en el Capígulo 2, son utilizados para representar datos de forma estructurada y significativa, aunque la adopción de estos estándares sigue siendo mayoritariamente en el ámbito académico. Existen iniciativas para su uso en entornos industriales, aunque todavía requiere de mayor impulso y de la generación de casos de uso reales.

- Escala y rendimiento: A medida que la cantidad de datos en la web continúa creciendo exponencialmente, es necesario abordar problemas de escalabilidad y rendimiento en la gestión de datos semánticos. Consultas complejas sobre grandes conjuntos de datos pueden ser costosas en términos de recursos computacionales.

- Integración de datos: La Web Semántica tiene como objetivo conectar datos de diversas fuentes y dominios, pero la integración de datos de manera efectiva sigue siendo un reto dada la diversidad de dominios y de fuentes. Esto implica mapear y reconciliar datos de diferentes ontologías y vocabularios.

- Calidad de los datos: Garantizar la calidad de los datos semánticos es fundamental. Los datos incorrectos o inconsistentes pueden llevar a interpretaciones erróneas y decisiones inapropiadas. La validación y la limpieza de datos son procesos críticos.

- Privacidad y seguridad: Con la creciente cantidad de datos semánticos disponibles en la web, surgen preocupaciones sobre la privacidad y la seguridad. La exposición de información sensible en la web y en conjuntos de datos accesibles puede ser un riesgo, por lo que es esencial desarrollar técnicas de anonimización y control de acceso.

- Interoperabilidad: Asegurar que sistemas y aplicaciones basados datos puedan interoperar de manera efectiva con otras tecnologías es un desafío constante. La Web Semántica puede contribuir en la conciliación semántica de datos y de formatos interoprables.

- Actualización y evolución de ontologías: Las ontologías, que definen la estructura y el significado de los datos en la Web Semántica, deben evolucionar para reflejar cambios en el mundo real. Mantener ontologías actualizadas y relevantes es un desafío constante.

Debido a estos retos, las tecnologías de la Web Semántica sigue siendo una área de investigación y desarrollo activa, que está tomando impulso en los últimos años gracias a la reciente generación de los Espacios Europeos de Datos[1], en los que pueden hacer contribuciones significativas al proporcionar un marco para la organización, el intercambio y la comprensión de datos de manera más efectiva. Entre algunos de los posibles aportes de la Web Semántica a los espacios de datos, cabe mencionar, por ejemplo:

- Enriquecimiento de datos: Las tecnologías semánticas permiten enriquecer los datos existentes al agregar metadatos y relaciones semánticas. Esto es especialmente útil para mejorar la calidad y la coherencia de los datos.

- Descubrimiento y acceso de datos: La Web Semántica proporciona herramientas para facilitar el descubrimiento de datos, lo que es fundamental para acceder a la información necesaria en un contexto europeo diverso y distribuido.

[1]Sitio web de European Data Spaces (última visita 19-10-2023)`https://dataspaces.info/common-european-data-spaces`

- Gestión de datos abiertos: La Web Semántica puede respaldar la gestión y la publicación de datos abiertos de manera coherente y con una semántica común. Esto es esencial para cumplir con iniciativas de datos abiertos en toda Europa.
- Integración de información geoespacial: La combinación de datos geoespaciales con datos semánticos puede ser especialmente relevante en contextos europeos, donde la ubicación y la geografía son fundamentales para muchas aplicaciones, como la planificación urbana, la agricultura, la gestión de recursos naturales, etc.
- Gobernanza de datos: facilitar la creación de políticas y estándares de gobernanza de datos compartidos en toda Europa. Esto es fundamental para garantizar la calidad, la seguridad y la confidencialidad de los datos en espacios de datos transfronterizos.
- Aplicaciones para ciudadanos y empresas: respaldar el desarrollo de aplicaciones y servicios que beneficien a ciudadanos y empresas europeos al proporcionar acceso a datos más ricos y contextualmente relevantes.
- Colaboración y comunicación internacional y multi-idioma: mejorar la comunicación y la colaboración entre países y organizaciones en Europa al permitir una comprensión más precisa y una interpretación más consistente de los datos compartidos.
- Cumplimiento de regulaciones y normativas: En un entorno europeo con diversas regulaciones y normativas, la Web Semántica puede ayudar a rastrear y gestionar el cumplimiento de estas normativas mediante la clasificación y el etiquetado semántico de datos.

En definitiva, las tecnologías de la Web Semántica pueden desempeñar un papel crucial en la creación de espacios de datos europeos más coherentes, eficientes y colaborativos. De hecho, estas tecnologías facilitan la gestión de datos compartidos a través de fronteras y contribuyen a una mejor gobernanza de datos, la interoperabilidad y el acceso a información relevante para una amplia variedad de aplicaciones en toda Europa.

1.1. Motivación

El papel de la Web Semántica viene siendo cada vez más relevante debido a su capacidad para organizar y relacionar información de manera más efectiva, por lo que motiva la investigación en nuevos métodos y aplicaciones reales basados en esta tecnología. Entre estas aplicaciones podemos destacar, desde la búsqueda en Internet hasta la inteligencia artificial y la toma de decisiones empresariales.

En este sentido, cabe resaltar el empleo de grafos de conocimiento semánticos para representar información en una estructura de nodos y relaciones, lo que facilita la organización de datos complejos y su conexión en una red coherente. Esto es particularmente útil en la gestión de datos estructurados y no estructurados.

Igualmente, esta tecnología tiene un gran potencial como soporte a procesos de análisis de datos basados en métodos de Inteligencia Artificial (IA), ya que ayudan a modelar y representar el conocimiento en diversas aplicaciones, desde la visión por computador y toma de decisiones autónomas, hasta la interpretación de las salidas algorítmicas. En particular, dentro del Procesamiento del Lenguaje Natural (NLP), los grafos semánticos son esenciales, ya que ayudan a comprender la semántica de las palabras y las oraciones, con importantes aportes en aplicaciones como chatbots, asistentes virtuales y traducción automática. Además, siguiendo con esta línea, es importante explorar la posible relación entre los bots conversacionales basados en grandes modelos lingüísticos (LLM), como ChatGPT [2], y el concepto de Web Semántica. Esta simbiosis se puede dar por

[2]Sitio Web de ChatGPT`https://chat.openai.com/`

varias razones: en primer lugar, los LLM proporcionan una base para un potente procesamiento del lenguaje natural basado en su comprensión de la sintaxis y la semántica de las frases; por ejemplo, comprenden la semántica subyacente de múltiples variaciones de la misma frase; En segundo lugar, la Web Semántica se inició como una variación de la web construida explícitamente a partir de frases que utilizan hipervínculos para expresar la sintaxis y la semántica de la frase de forma computable por la máquina; el efecto es un conjunto ilimitado de datos estructurados que manifiesta un grafo global de relaciones entre entidades (en lugar de una red), que comprende una semántica de tipo relación entre entidades computable por la máquina.

De manera transversal, el empleo de estas tecnologías ya es una realidad en la automatización de procesos empresariales y científicos basados en datos, al servir de hilo conductor para modelar workflows, recursos y dependencias. Esto mejora la eficiencia, da soporte a la trazabilidad del dato y reduce los errores o inconsistencias a la hora de conectar componentes de procesamiento y análisis. La plataforma TITAN [4] supone un avance en este sentido, pues ya está siendo utilizada en proyectos de ciencia ciudadana en el ámbito europeo.

Todas estas razones sirven de inspiración y motivación de la presente Tesis Doctoral, poniendo de relieve la necesidad de ir un paso más allá en la generación de nuevas ontologías y modelos semánticos. La idea es extraer experiencias y conclusiones comunes a la hora de explorar su potencial en aplicaciones reales en dominios de conocimiento heterogéneos, así como su capacidad para representar el conocimiento de manera estructurada y comprender las relaciones entre conceptos, de manera que de soporte a procesos de inteligencia artificial en el análisis de datos. En base a ésto, se formulan las siguientes preguntas de investigación, que nos han guiado en el proceso de modelado, experimentación y validación de las propuestas científicas en diversos casos de uso.

Preguntas de investigación

- **Q1**: ¿Cómo se pueden aprovechar los modelos semánticos y los grafos de conocimiento para mejorar la integración, armonización y estandarización de datos?
- **Q2**: ¿Cómo se pueden combinar los modelos semánticos y grafos de conocimiento con otras tecnologías emergentes, como la inteligencia artificial y el aprendizaje automático, para mejorar la calidad y el valor del análisis de datos en entornos de e-learning?
- **Q3**: ¿Cómo pueden las técnicas de inferencia semántica automatizar procesos de conciliación de datos y determinación de la solvencia financiera en servicios de banca abierta guiados por datos?
- **Q4**: ¿Cómo se pueden utilizar los modelos semánticos y los grafos de conocimiento para mejorar la seguridad y la privacidad en las bases de datos NoSQL?

Además, estas principales preguntas de investigación pueden asociarse a diversos retos que se presentan en problemas del mundo real en diversas áreas.

Retos de investigación

- **R1**: Necesidad de modelos conceptuales formales. Desarrollar y crear modelos semánticos y grafos de conocimiento precisos y confiables para diferentes áreas temáticas, orientados a la integración efectiva de múltiples fuentes de datos, la armonización de datos relacionados o la estandarización de reglas aplicables en diferentes entornos.
- **R2**: Técnicas de combinación de grafos de conocimiento y algoritmos de Machine Learning. Desarrollo de técnicas que exploten la semántica de dominio y los grafos de conocimiento como entrada a los algoritmos de aprendizaje automático y que aporten valor añadido al análisis matemático de datos.

- **R3**: Explotación avanzada de la semántica de dominio en servicios de banca abierta basados en la normativa PSD2. Es necesario definir metodologías basadas en semántica y razonamiento que permitan armonizar datos bancarios con el objetivo de mejorar las técnicas de clasificación de usuarios y determinar su solvencia económica.

- **R4**: Necesidad de métodos estándares para definir reglas de seguridad en bases de datos NoSQL. En la actualidad las bases de datos NoSQL carecen de una metodología estándar para la definición y creación de políticas de seguridad y privacidad en la fase de diseño. Es necesario explotar el uso de semántica para abstraer la definición y validación de dichas políticas de la implementación final.

1.2. Objetivos y fases

Esta Tesis Doctoral se centra en la integración de diferentes fuentes de información y la mejora del análisis de datos, de cara a la toma de decisiones a través del uso de técnicas emergentes de Web Semántica. Para lograr este objetivo general, se propone un marco de trabajo común mediante la generación de modelos y ontologías, que faciliten la integración semántica de datos y su representación en forma de grafo, lo que facilita el análisis de datos y la extracción de conocimiento.

En concreto, se destacan a continuación una serie de objetivos específicos alineados con el objetivo principal, aunque poniendo enfoque en los dominios de aplicación seleccionados:

- **Objetivo 1: Diseñar e implementar ontologías y grafos de conocimiento en los dominios específicos, que permitan implementar soluciones de integración, armonización y estandarización de datos,** así como la explotación de sus capacidades de razonamiento y análisis de datos en procesos de inteligencia artificial.

- **Objetivo 2: Siguiendo un modelo semántico, alimentar mediante un grafo de conocimiento una serie de algoritmos de Machine Learning capaces de analizar patrones de interacción implícitos en e-Learning Management Systems (LMSs).** El grafo de conocimiento generado contemplará la integración de diversas fuentes de datos reales y académicas sobre la interacción de estudiantes y profesores en comunidades universitarias.

- **Objetivo 3: Definir un grafo de conocimiento que alinee transacciones de banca abierta PSD2 con información de facturas para realizar estudios de solvencia bancaria.** Asimismo, se pretende obtener un sistema de reglas semánticas para mostrar cómo la clasificación de solvencia financiera de las entidades cliente y las sugerencias de conceptos de transacción, pueden inferirse a partir del modelo semántico propuesto.

- **Objetivo 4: Definir un marco de trabajo guiado por ontología para diseñar bases de datos seguras basadas en grafos, que facilite la migración rápida de las reglas de seguridad derivando en medidas específicas para cada tecnología subyacente.** Se pretende aportar a los diseñadores de bases de datos de métodos para comprobar mediante el razonamiento ontológico si las reglas de seguridad son coherentes.

En general, las fases para alcanzar estos objetivos incluyen: la definición de los requisitos de información del dominio, la selección de las fuentes de datos relevantes, el diseño y desarrollo de modelos semánticos y ontologías, la adaptación de técnicas de análisis de datos avanzadas y la validación de los resultados mediante casos de uso específicos. Esto último es importante para demostrar la utilidad y eficacia de los modelos y marcos de trabajo propuestos en situaciones reales.

En particular, se exponen las fases de la metodología común seguida en esta Tesis Doctoral, dando homogeneidad a la concepción de cada caso de diseño, en los dominios específicos de aplicación:

- **Fase 1: Identificación de las fuentes de información relevantes,** considerando tanto otras ontologías existentes relacionadas, como conjuntos de datos y esecificación de requisitos.

- **Fase 2: Diseño del modelo semántico.** Esta fase implica la definición de los conceptos y relaciones que se utilizarán para representar la información y la estructura de estos elementos en una ontología.

- **Fase 2: Implementación del modelo semántico.** En esta fase se implementa el modelo semántico mediante el uso de herramientas específicas para la creación de ontologías. Se contempla el enlazado con ontologías relacionadas existentes y la modificación, en su caso, para la adaptación a los nuevos requisitos.

- **Fase 3: Poblado el grafo de conocimiento y métodos de persistencia.** En esta fase se procede a integrar las diferentes fuentes de información identificadas en la primera fase. Esto puede implica la realización de tareas de extracción, transformación y carga, mappings de datos e integración, para adaptarlos al modelo semántico definido. Se establece también la base de datos para almacenado del grafo y servicios EndPoint para consulta.

- **Fase 4: Validación del modelo semántico.** Se procede a validar el modelo semántico desarrollado mediante casos de uso específicos. Esta fase puede incluir la evaluación de la calidad de la información integrada, mediante consultas prediseñadas, la verificación de la consistencia del modelo, y la evaluación del rendimiento de las herramientas y aplicaciones que utilizan el modelo semántico. Tras esta evaluación, se podrá regresar a la Fase 2 para la reimplementación y refinado del modelo.

- **Fase 5: Selección y aplicación de técnicas de análisis de datos avanzadas.** Esta fase conlleva la selección y aplicación las técnicas de análisis de datos avanzadas más adecuadas para el dominio de estudio y los datos integrados. Esto implica la exploración de los datos para identificar patrones y relaciones, la generación de las consultas específicas para el nutrido de datos, la selección y ajuste de modelos de predicción, la evaluación de la precisión y robustez de los modelos.

- **Fase 6: Toma de decisiones informadas.** En esta fase se utiliza la información obtenida a partir de la integración de las fuentes de información y el análisis de datos avanzado para tomar decisiones informadas en el dominio de estudio. Esto puede implicar la identificación de patrones y tendencias relevantes, la evaluación de diferentes escenarios y opciones, junto con la selección de la mejor solución en función de los objetivos y restricciones del problema en cuestión.

- **Fase 7: Evaluación de los resultados y mejora continua.** Finalmente, en esta fase se evalúan los resultados obtenidos a partir de la integración de las fuentes de información y el uso de técnicas de análisis de datos avanzadas, y se identifican áreas de mejora y oportunidades de aprendizaje para mejorar la toma de decisiones y la precisión de las predicciones en futuros proyectos o situaciones similares.

1.3. Contribuciones y publicaciones científicas

Fruto de la labor realizada en esta Tesis, se han generado una serie de ontologías derivadas de las investigaciones realizadas (e-LION [3], OBO [4] y OntoSecurityGraphDB [5]), así como 3 publicaciones científicas de impacto, en las cuales se recogen las contribuciones específicas detalladas en los Capítulos 3, 4 y 5. Estas publicaciones han sido evaluadas y aceptadas en revistas científicas internacionales indexadas en el Journal of Citation Report (JCR), estando dos de ellas en cuartil Q1 (categorías de ciencias de la computación) y otra en Q2 (categoría de ingeniería multidisciplinar). Además, se realizan sendas presentaciones en las Jornadas de Ingeniería del Software y Bases de Datos (JISBD), en sus ediciones de 2022 y 2023.

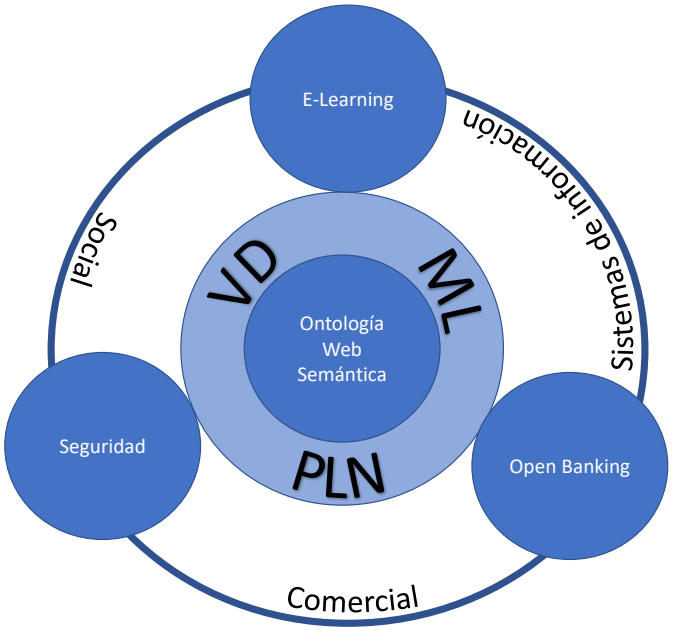

Figura 1.1: Diagrama conceptual de los aspectos que abarca esta Tesis.

A continuación se detallan estas publicaciones científicas, si bien préviamente, a modo de ilustración general de los temas abordados en esta Tesis, se muestra en la Figura 1.1 un diagrama conceptual de los aspectos principales y sus relaciones, haciendo referencia a las contribuciones científicas. Dentro de su alcance, se contempla la propuesta de ontologías y modelos semánticos en conjunción con técnicas de Machine Learning (ML), de Procesamiento de Lenguaje Natural (PLN), y de Visualización de Datos (VD), en dominios de conocimiento específicos de e-Learning, Open Banking y Seguridad. Estos dominios de aplicación fueron seleccionados a su vez por aporte en el aspecto social, comercial y tecnológico.

[3] https://ontologies.khaos.uma.es/e-lion
[4] https://ontologies.khaos.uma.es/obo
[5] https://proyectoaether.github.io/OntoSecurityGraphDB/index-en.html

Publicaciones en revistas (indexadas en JCR) que avalan esta Tesis Doctoral

1. Manuel Paneque, María del Mar Roldán-García, José García-Nieto, ***e-LION: Data Integration Semantic Model to Enhance Predictive Analytics in e-Learning***, en la revista *Expert Systems With Applications* (Q1, En la categoría *Computer Science, Artificial Intelligence*, Posición: 21/145, Factor de impacto: 8.665, DOI: `https://doi.org/10.1016/j.eswa.2022.118892`).

   En el trabajo "e-LION: Data Integration Semantic Model to Enhance Predictive Analytics in e-Learning", se propone un enfoque innovador en el análisis predictivo en plataformas de aprendizaje electrónico. Utiliza un modelo semántico de integración de datos basado en una ontología, llamada e-LION, para integrar y analizar datos de diversas fuentes en el entorno del aprendizaje online. Este enfoque semántico es novedoso ya que permite una comprensión más profunda y precisa de los datos, lo que conduce a mejores predicciones sobre el rendimiento y la motivación de los estudiantes. La implementación del modelo e-LION tiene como objetivo mejorar la efectividad del aprendizaje electrónico a través de una mejor comprensión de los factores que afectan el rendimiento y la motivación de los estudiantes.

2. Manuel Paneque, María del Mar Roldán-García, José García-Nieto, ***A Semantic Model for Enhancing Data-Driven Open Banking Services***, en la revista *Applied Sciences* (Q2, En la categoría *Engineering, Multidisciplinary*, Rank: 39/92, Impact Factor: 2.838, DOI: `https://doi.org/10.3390/app13031447`).

   En el trabajo "A Semantic Model for Enhancing Data-Driven Open Banking Services" se centra en mejorar los servicios de banca abierta basados en datos. Propone un modelo semántico para integrar y analizar datos de diferentes fuentes en el entorno de banca abierta. Este modelo permite una comprensión más profunda y precisa de los datos, lo que conduce a mejores decisiones de negocios y mejores servicios para los clientes. La implementación del modelo semántico tiene como objetivo mejorar la eficiencia y la efectividad de los servicios de banca abierta.

3. Manuel Paneque, María del Mar Roldán-García, Carlos Blanco, Alejandro Maté, David G. Rosado, Juan Trujillo, ***An Ontology-based Secure Design Framework for Graph-based Databases.*** en la revista *Computer Standards & Interfaces*, (Q1, en la categoría *Computer Science, Sotware Engineering*, Rank: 24/110, Impact Factor: 3.721, DOI: `https://doi.org/10.1016/j.csi.2023.103801`).

   En el trabajo "An Ontology-based Secure Design Framework for Graph-based Databases" se aborda la importancia de la seguridad en las bases de datos basadas en grafos. En el mundo digital actual, las bases de datos son un recurso crítico para la mayoría de las organizaciones, y la seguridad de estos datos es esencial para garantizar la privacidad y la confidencialidad de la información. Sin embargo, las bases de datos basadas en grafos presentan desafíos únicos en cuanto a seguridad debido a su estructura altamente conectada y su capacidad para representar relaciones complejas entre entidades. Se presenta pues un marco novedoso de diseño de seguridad basado en ontología para abordar estos desafíos y proporcionar una solución efectiva el diseño de la seguridad de las bases de datos basadas en grafos y su consistencia. Este marco utiliza conceptos de ontología para modelar la estructura de la base de datos y definir los derechos de acceso y control de seguridad apropiados. Además, también se verifica el modelo semántico con el modelado de la seguridad en una base de datos y generando de forma automática el código para implementar el modelo y la seguridad en diferentes sistemas gestores de bases de datos orientadas a grafos.

Publicaciones en revistas (indexadas en JCR) adicionales

- Antonio Benítez-Hidalgo, Cristóbal Barba-González, José García-Nieto, Pedro Gutiérrez-Moncayo, Manuel Paneque, Antonio J. Nebro, María del Mar Roldán-García, José F. Aldana-Montes, Ismael Navas-Delgado, ***TITAN: A knowledge-based platform for Big Data workflow management.*** en la revista *Knowledge-Based Systems*, (Q1, en la categoría *Computer Science, Artificial Intelligence*, Rank: 21/145, Impact Factor: 8.139, DOI: `https://doi.org/10.1016/j.knosys.2021.107489`).

  En este trabajo se presenta una plataforma innovadora para la gestión de flujos de trabajo de Big Data. La plataforma TITAN utiliza un enfoque basado en conocimiento y semántica para automatizar y optimizar los procesos de trabajo. Además, facilita al usuario en el diseño del flujo, guiándolo en cada paso para asegurarse de que se cumplan los objetivos de los procesos de trabajo. La semántica es fundamental para garantizar la verificación de los flujos de trabajo y asegurar la integridad de los datos a lo largo del proceso. La plataforma también permite la integración y el análisis de datos de diferentes fuentes, mejorando la precisión y eficiencia de la toma de decisiones basadas en datos. La implementación de TITAN tiene como objetivo mejorar la eficiencia y efectividad de la gestión de flujos de trabajo de Big Data, la confianza en la verificación e integridad de los datos, así como facilitar al usuario en el diseño eficiente de flujos de trabajo.

Publicaciones en congreso

Se realizan 2 contribuciones en las Jornadas de Ingeniería del Software y Bases de Datos (JISBD) en sus ediciones de 2022 y 2023. Estas contribuciones se presentan en las respectivas sesiones especiales de "Artículo Relevante", en las que se describen las propuestas científicas desarrolladas en los artículos anteriores.

1.4. Estructura del documento

Esta Tesis doctoral se organiza en una estructura de 6 capítulos que abarcan diversos casos de estudio en los que se aplica las tecnologías de la web semántica. Esta aplicación se enfocada en la integración de datos y el análisis avanzado de los datos enlazados. En primer lugar, se dedica el Capítulo 2 a la presentación detallada de los conceptos y tecnologías esenciales. Entre ellas se incluyen la Web Semántica, *"Machine Learning"*, Sistemas de *"E-learning"*, *"Open Banking"* y la Seguridad en Bases de Datos Orientadas a Grafos. Este análisis aborda los principios, aplicaciones y relevancia de estas tecnologías en el contexto de la tesis.

A continuación, se presentan tres capítulos dedicados a modelos semánticos específicos y su aplicación en dominios particulares. El Capítulo 3 describe un modelo semántico de integración de datos diseñado para mejorar el análisis predictivo en plataformas de *"E-Learning"*. El Capítulo 4, detalla un modelo semántico para la integración de datos como soporte a la directiva PSD2 de banca abierta, abordando la necesidad de integración semántica en el contexto de la banca abierta enlazada. Por último, se expone en el Capítulo 5 un marco de trabajo centrado en la gestión de la seguridad en bases de datos orientadas a grafos, proporcionando una estructura para garantizar la seguridad de los datos en este tipo de bases de datos.

Finalmente, la Tesis concluye con el Capítulo 6 que resume las principales contribuciones realizadas y plantea las direcciones a seguir en futuras investigaciones. Se analizan las limitaciones de la investigación actual y se esbozan posibles vías para ampliar y mejorar los conceptos presentados a lo largo de la Tesis doctoral. Este capítulo de conclusiones y trabajo futuro cierra la investigación de manera integral y proporciona una visión completa del camino recorrido y las oportunidades por explorar en el futuro.

Capítulo 2

Contexto y fundamentos

En este capítulo, se explica brevemente diversas tecnologías y conceptos fundamentales que impulsan la creación de grafos de conocimiento. Estas tecnologías sirven como herramientas para representar y relacionar información de manera semántica. Entre las tecnologías usadas en esta Tesis se encuentran la Web Semántica y el *"Machine Learning"*, aplicadas en dominios como el *"e-Learning"*, *"Open Banking"* y la *"seguridad en bases de datos orientadas a grafos"*.

Finalmente, se procede a introducir cada uno de los dominios de estudio con el objetivo de comprender plenamente qué tipo de datos se ven involucrados en cada uno de ellos. En el ámbito del *"e-Learning"*, nos sumergiremos en una gran cantidad de información educativa, que abarca desde la descripción de recursos didácticos hasta los registros de las interacciones de los usuarios con dichos recursos y las calificaciones obtenidas. En el dominio del *"Open Banking"*, exploramos los datos financieros que fluyen a través de transacciones, cuentas bancarias y perfilaremos a los clientes, desvelando así patrones y coincidencias clave. Por último, y muy en relación con las anteriores, nos centramos en el ámbito de la *"seguridad en bases de datos orientadas a grafos"*, donde se aborda la protección de datos sensibles y la gestión de acceso basada en roles, encarando y dando solución a cuestiones cruciales para gestionar la seguridad en el paradigma de información masiva y distribuida. A medida que nos sumergimos en cada dominio, se despliega un panorama interoperable, extensible y dinámico de datos que, con la aplicación de las tecnologías adecuadas, nos permite desvelar conocimientos profundos y relaciones ocultas, enriqueciendo así nuestra comprensión del dominio de estudio en su conjunto.

2.1. Tecnologías de la Web Semántica

En esta sección, comenzamos introduciendo conceptos relacionados con las tecnologías de la Web Semántica con el fin de facilitar la lectura de este manuscrito, dándole así autocontenido suficiente para su seguimiento. Estas tecnologías proporcionan un marco conceptual y tecnológico para enriquecer la información con metadatos semánticos, lo que facilita su interpretación automática por las máquinas. Mediante estándares como RDF del inglés *("Resource Description Framework")* y OWL *("Web Ontology Language")*, se puede representar la semántica de los datos de forma estructurada, estableciendo relaciones entre entidades y permitiendo su reutilización en diferentes contextos.

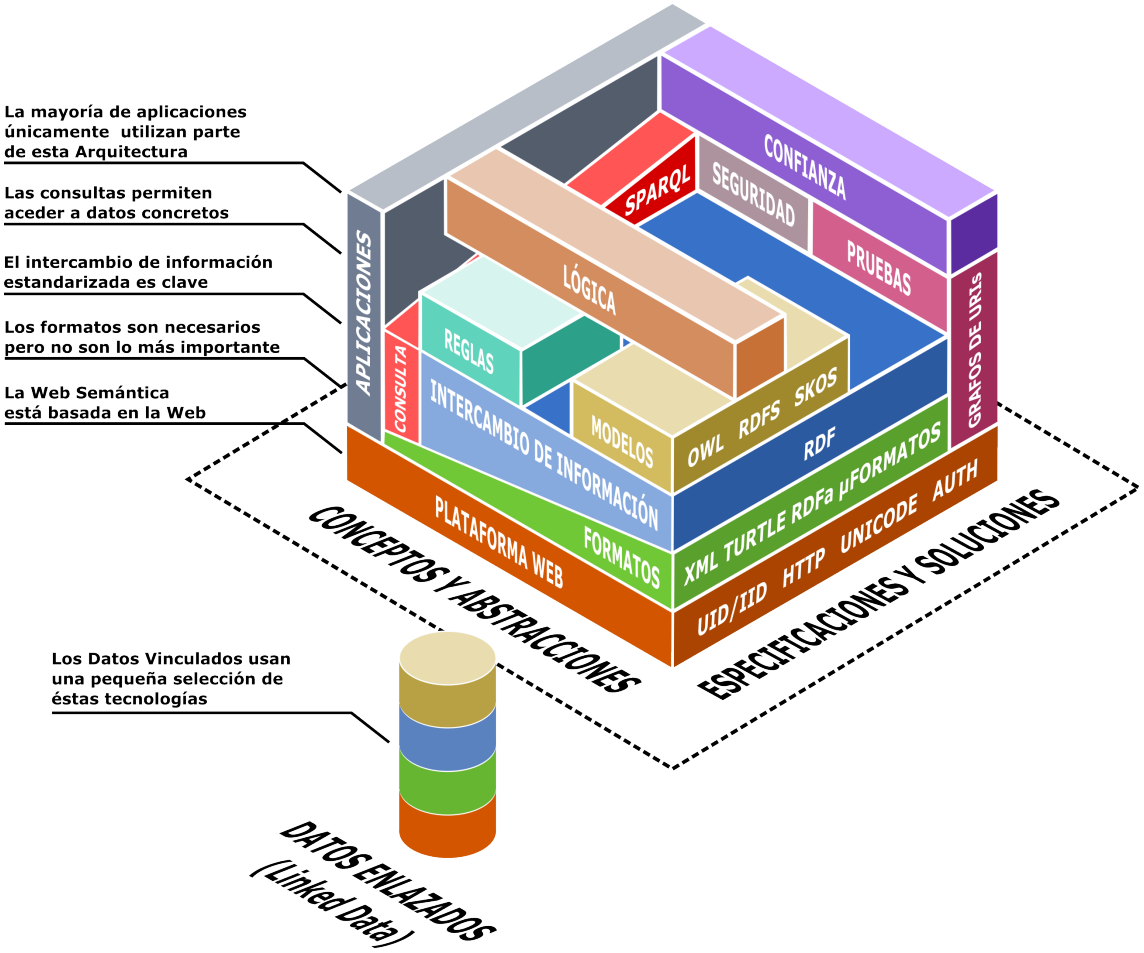

Figura 2.1: Pila de tecnologías de la Web Semántica. Fuente [6].

Ontología. De acuerdo con referencias iniciales en este área [7] y [8], una ontología proporciona una representación formal del mundo real. Las ontologías son una poderosa herramienta para representar el conocimiento en un dominio concreto. Las ontologías definen una descripción explícita de los conceptos, relaciones, atributos y restricciones en un campo. Se utilizan como marco de comunicación entre personas y organizaciones al proporcionar una terminología consensuada sobre un dominio. Una de las principales características de las ontologías es que permiten el ra-

zonamiento automático sobre los datos. Las ontologías forman parte de la pila de estándares de la web semántica del W3C (Consorcio de la *"World Wide Web"*). Una ontología y un conjunto de individuos forman una base de conocimiento, que proporciona servicios para facilitar la interoperabilidad entre múltiples sistemas y bases de datos heterogéneas.

RDF. *"Resource Description Framework"* [9] es una recomendación del W3C que define un lenguaje para describir recursos en la web. RDF describe los recursos como tripletas, formadas por un sujeto, un predicado y un objeto. RDF puede representar individuos ontológicos en forma de grafo. El término grafo de conocimiento ha surgido recientemente para referirse a descripciones interconectadas de entidades, codificando al mismo tiempo la semántica subyacente a la terminología utilizada (se incluye una definición más adelante). Por lo tanto, en el contexto de este manuscrito, los grafos RDF son grafos de conocimiento. RDF *"Schema"* (RDFS) [10] describe los vocabularios utilizados en las descripciones RDF.

Existen diferentes serializaciones estandarizadas para representar datos RDF según convenga en la aplicación de destino:

- **RDF/XML** [1]: Es una serialización basada en XML que permite representar datos RDF. Utiliza etiquetas XML para representar los sujetos, predicados y objetos, y permite la incorporación de namespaces para definir individuos y vocabularios.
- **Turtle** [2]: Del inglés *"Terse RDF Triple Language"* es una serialización legible por humanos y compacta para RDF. Utiliza una sintaxis sencilla basada en texto que utiliza prefijos para reducir la repetición de información.
- **N-Triples** [3]: Es una serialización muy simple y minimalista para RDF. Utiliza una estructura de líneas de texto que representa cada tripleta RDF en su forma más básica, sin abreviaciones o redundancias.
- **JSON-LD** [4]: Es una serialización basada en JSON. Permite representar los datos RDF utilizando la notación de objetos de JSON, lo que facilita su uso e integración con aplicaciones web y servicios.
- **RDFa** [5]: Del inglés *"RDF in Attributes"* es una técnica para incrustar metadatos RDF en documentos HTML. Permite la inserción de atributos HTML para enriquecer un documento HTML con datos RDF.

Las tripletas son una estructura utilizada en el lenguaje RDF para representar información de manera sencilla y estructurada. Estas tripletas constan de tres elementos principales: el sujeto, el predicado y el objeto. El sujeto se refiere a la entidad o recurso al cual se le está atribuyendo información. El predicado establece una relación o propiedad entre el sujeto y el objeto. El objeto representa la entidad asociada al sujeto.

Es importante destacar que estas tripletas permiten establecer una dirección o flujo de información desde el sujeto hacia el objeto. Esto significa que la relación expresada por el predicado se aplica desde el sujeto hacia el objeto. Sin embargo, es posible que un mismo recurso aparezca tanto como sujeto en una tripleta y como objeto en otra, lo cual nos brinda la posibilidad de definir conexiones y relaciones entre diferentes tripletas.

[1] RDF/XML `https://www.w3.org/TR/rdf-syntax-grammar/`
[2] N-Triples `https://www.w3.org/TR/n-triples/`
[3] N-Triples `https://www.w3.org/TR/n-triples/`
[4] JSON-LD `https://www.w3.org/TR/json-ld11/`
[5] RDFa `https://www.w3.org/TR/rdfa-primer/`

Para separar los términos de una tripleta, se pueden utilizar espacios en blanco o tabuladores. Además, cada tripleta se finaliza con un punto (.) y un salto de línea para indicar su final.

Los términos utilizados en las tripletas están compuestos por IRI, estas IRIs se encierran entre los símbolos "<" y ">". Por ejemplo, una IRI podría ser "http://w3id.org/sujeto1".

Fragmento de código 2.1: Sintaxis para representar tripletas.

```
<http://w3id.org/sujeto1> <http://w3id.org/predicado1> <http://w3id.org/objeto1> .
<http://w3id.org/sujeto1> <http://w3id.org/propiedad1>
"valor"^^<http://www.w3.org/2001/XMLSchema#Tipo> .
```

OWL. El lenguaje de definición de ontologías web (*"Ontology Web Language"*) se utiliza para especificar ontologías en la web, este lenguaje amplía RDF y RDFS, pero añadiendo vocabulario. OWL es equivalente a una DL (lógica de descripción) muy expresiva, donde una ontología corresponde a una *"Tbox"* (Caja de Terminología) [11]. Esta equivalencia permite al lenguaje explotar los resultados de la investigación mediante la lógica de descripción definida. OWL proporciona dos sublenguajes: *"OWL Lite"*, para aplicaciones sencillas, y OWL-DL, que representa el subconjunto de lenguajes equivalentes a la lógica de descripción, cuyos mecanismos de razonamiento son bastante complejos. OWL-DL es una descripción sintáctica que proporciona la máxima expresividad al tiempo que conserva la completitud computacional y la decidibilidad [12].

El lenguaje completo se denomina OWL Full. La versión 2 de OWL se publicó en 2009, incluyendo nuevas características y definiendo tres nuevos perfiles, OWL 2 EL, OWL 2 QL y OWL 2 RL [13], y una nueva sintaxis (OWL 2 *"Manchester Syntax"*). Además, se han relajado algunas de las restricciones aplicables a *"OWL DL"* y, como resultado, el conjunto de grafos RDF que pueden manejar los razonadores de lógica descriptiva es ligeramente mayor en OWL 2. Teniendo en cuenta que OWL 2 es un lenguaje expresivo popular que incluye un mayor poder expresivo para las propiedades, soporte ampliado para los tipos de datos, capacidades simples de metamodelado, capacidades ampliadas de anotación y claves, lo utilizamos para definir nuestra ontología. Además, OWL 2 especifica una correspondencia precisa entre estructuras ontológicas y grafos RDF, teniendo una correspondencia explícitamente especificada entre los grafos RDF y las estructuras ontológicas.

Tabla 2.1: Sintaxis básica del lenguaje *"OWL-DL"* utilizada para definir formalmente las ontologías propuestas. La Tabla está organizada por Operadores (O), Restricciones (R) y Axiomas de Clase (A).

	Sintaxis Abstracta	Sintaxis en DL
O	$intersección(C_1, C_2, \cdots, C_n)$	$C_1 \sqcap C_2 \sqcap \cdots C_n$
	$unión(C_1, C_2, \cdots, C_n)$	$C_1 \sqcup C_2 \sqcup \cdots C_n$
R	para al menos un valor V de C	$\exists V.C$
	para todos los valores V de C	$\forall V.C$
	R es Simétrica	$R \equiv R^-$
A	$A\ parcial(C_1, C_2, \cdots, C_n)$	$A \sqsubseteq C_1 \sqcap C_2 \sqcap \cdots C_n$
	$A\ completo(C_1, C_2, \cdots, C_n)$	$A \equiv C_1 \sqcap C_2 \sqcap \cdots C_n$

Existen diferentes tipos de sintaxis para definir ontologías, pero la más usada es OWL/XML. Las clases en el contexto de OWL son una forma de agrupar recursos con características similares. Se puede pensar en ellas como una jerarquía de clases o conjuntos de elementos relacionados.

Cuando se define una clase en OWL, se emplea una sintaxis específica basada en XML. Además, utilizamos el prefijo "owl" seguido de "Class" para indicar que se trata de una clase. Así, establecemos que la entidad que se está describiendo pertenece a la categoría de clases en el lenguaje OWL.

La referencia a la URI de la clase en RDF es un enlace único que identifica de manera única a esa clase en el contexto de la web semántica.

Fragmento de código 2.2: Sintaxis para representar clases en OWL.

```
<owl:Class rdf:about="http://w3id.org/OntologyIdentifier/NombreClase">
```

Las propiedades de objeto son un concepto utilizado en el ámbito de la web semántica para establecer una conexión entre dos clases. Para indicar que se trata de una propiedad de objeto, utilizamos el prefijo "owl" segido de "ObjectProperty" y la referencia a la URI de la propiedad que queremos definir.

Además, es importante especificar el dominio y el rango de la relación. El dominio se refiere a la clase a la que pertenece la clase desde la cual se inicia la relación. Para indicar el dominio, utilizamos el prefijo "rdfs:domain" seguido de la URI de la clase correspondiente. Por otro lado, el rango se refiere a la clase con la que termina la relación. Para indicar el rango, utilizamos el prefijo "rdfs:range" seguido de la URI de la clase correspondiente. Esto nos permite definir qué tipo de instancias de esas clases pueden estar relacionadas por la propiedad de objeto definida.

Fragmento de código 2.3: Sintaxis para representar propiedades de objeto en OWL.

```
<owl:ObjectProperty rdf:about="http://w3id.org/OntologyIdentifier/NombrePropiedadObjeto">
    <rdfs:domain rdf:resource="http://w3id.org/OntologyIdentifier/ClaseDominio">
    <rdfs:range rdf:resource="http://w3id.org/OntologyIdentifier/ClaseRango">
</owl:ObjectProperty>
```

Las propiedades de datos son utilizadas para representar la relación de una clase con tipos de datos simples. En OWL, se emplean la mayoría de los tipos de datos definidos en el esquema XML. Estos tipos de datos se referencian mediante la URI correspondiente al tipo de dato específico.

La sintaxis para declarar las propiedades de objetos se realiza mediante la utilización del prefijo "owl:DatatypeProperty" que indica que se trata de una propiedad de datos, y a continuación se especifica la URI del nombre asignado a dicha propiedad. Para indicar el dominio de la relación, se utiliza el prefijo "rdfs:domain" seguido de la URI de la clase que pertenece al dominio.

Asimismo, para indicar el rango se emplea el prefijo rdfs:range", seguido de la URI del tipo primitivo definido en el esquema XML, como por ejemplo: `http://www.w3.org/2001/XMLSchema#TipoDato`.

Fragmento de código 2.4: Sintaxis para representar propiedades de datos en OWL.

```
<owl:DatatypeProperty rdf:about="http://w3id.org/OntologyIdentifier/NombrePropiedadDatos">
    <rdfs:domain rdf:resource="http://w3id.org/OntologyIdentifier/ClaseDominio">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#TipoDato">
</owl:DatatypeProperty>
```

SPARQL. Es un lenguaje de consulta para acceder fácilmente a almacenes RDF. Es el lenguaje de consulta recomendado por el W3C [14] para trabajar con grafos RDF [15], soportando consultas y fuentes de datos web identificadas por URIs (Identificador Uniforme de Recursos).

Este lenguaje admite operadores y cláusulas comunes en otros lenguajes de consulta, como *"SELECT"*, *"COUNT"*, *"FILTER"*, *"GROUP BY"*, *"LIMIT"*, *"DESCRIBE"*, *"DISTINCT"*, entre otros. Además, incluye la cláusula *"ASK"*, que devuelve un valor booleano si existe algún resultado coincidente con el patrón especificado.

Además de las cláusulas y operadores mencionados anteriormente, SPARQL también ofrece soporte para realizar operaciones de *"JOIN"* mediante el paradigma basado en patrones de coincidencias de tripletas. Esto es especialmente útil cuando se trabaja con múltiples grafos RDF que

contienen datos interconectados. La interconexión de diferentes fuentes de datos distribuidos que soporten SPARQL se pueden especificar con la cláusula *"SERVICE"*.

Al combinar la claúsula *"SERVICE"* y el paradigma de patrón de coincidencia de tripletas, SPARQL se convierte en un lenguaje potente para realizar consultas complejas y distribuidas para obtener información detallada de grafos RDF. Permite expresar relaciones sofisticadas entre los datos almacenados en RDF y extraer resultados que satisfacen criterios específicos. Esto facilita la exploración y el análisis de grafos de conocimiento.

Fragmento de código 2.5: Ejemplo de consulta en SPARQL.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (COUNT(?submission) AS ?num_submissions) ?user_id ?course_id
WHERE{
    SERVICE <https://user-mud.khaos.uma.es/sparql> {
      ?user rdf:type elion:User .
      ?user elion:userId ?user_id .
      ?user elion:isEnrolled ?enroll .
      ?enroll elion:inCourse ?course .
      ?course rdf:type elion:Course .
      ?course elion:courseId ?course_id .
    }
    SERVICE <https://assignment-mud.khaos.uma.es/sparql> {
      ?submission elion:belongsUser ?user .
      ?submission elion:belongAssignment ?assigment .
      ?assigment elion:hasCourse ?course .
    }
}GROUP BY ?user_id ?course_id
```

SWRL. Es un lenguaje utilizado en el ámbito de la Web Semántica para representar y razonar sobre el conocimiento descrito por las ontologías [16]. Con SWRL, se pueden establecer reglas que constan de un antecedente y un consecuente. El antecedente establece las condiciones que deben cumplirse, y el consecuente especifica la afirmación que se deduce si se satisfacen las condiciones del antecedente. El antecedente se compone mediante proposiciones conectadas mediante la conjunción lógica "$\wedge$". Cada proposición describe una condición que debe cumplirse para que la regla se aplique. Esto permite inferir nuevas afirmaciones a partir de las existentes, facilitando el razonamiento automático y la toma de decisiones en aplicaciones basadas en las tecnologías de la Web Semántica. SWRL es una herramienta poderosa para modelar conocimiento y realizar razonamiento, lo que lo convierte en una herramienta valiosa en el desarrollo de sistemas inteligentes y la interconexión de datos en la era del Big Data.

SWRL cuenta con una extensión llamada SWRLB (Semantic Web Rule Language Built-ins) que proporciona una serie de funciones adicionales para manipular datos en reglas SWRL. Entre las funciones adicionales se encuentran operaciones y cálculos, como comparaciones numéricas, manipulación de cadenas de carácteres, funciones trigonométricas, ect. Estas funciones permiten una mayor expresividad y flexibilidad al definir reglas en SWRL.

En el Fragmento de Código 2.6, se muestra un ejemplo de una regla definida en SWRL. A continuación se explica cada una de las partes que la componen:

- prefijo:Clase1(?individuo) significa que el individuo (?individuo) debe pertenecer a la clase 'Clase1' definida con el prefijo 'prefijo'.
- prefijo:Propiedad1(?individuo, ?valorP1) establece que el individuo (?individuo) debe tener una propiedad 'Propiedad1' definida con el prefijo 'prefijo', y esa propiedad debe tener un valor (?valorP1).

- swrlb:greaterThanOrEqual(?valorP1, 50) especifica que el valor (?valorP1) debe ser mayor o igual a 50. Aquí se utiliza la función predefinida 'swrlb:greaterThanOrEqual' para realizar la comparación.
- Después del antecedente, usamos el operador '->' para indicar la consecuencia de la regla. Por lo tanto, prefijo:Propiedad2(?individuo, 'Alto') establece que si todas las condiciones del antecedente se cumplen, entonces se agrega una nueva propiedad 'Propiedad2' al individuo (?individuo), con el valor 'Alto'.

Fragmento de código 2.6: Regla básica definida en SWRL.

```
prefijo:Clase1(?individuo) ^ prefijo:Propiedad1(?individuo, ?valorP1)
^ swrlb:greaterThanOrEqual(?valorP1, 50)
-> prefijo:Propiedad2(?individuo, "Alto")
```

Grafo de Conocimiento. En el contexto de la representación del conocimiento y el razonamiento, un grafo de conocimiento (del inglés Knowledge Graph) es una base de conocimiento que utiliza un modelo de datos estructurado mediante grafos o cierta topología para integrar datos. Los grafos de conocimiento se utilizan para almacenar descripciones interconectadas de entidades: objetos, eventos, situaciones o conceptos abstractos; codificando al mismo tiempo la semántica subyacente a la terminología utilizada [17].

Desde el desarrollo de la Web Semántica, los grafos de conocimiento se asocian a menudo con proyectos de datos abiertos enlazados (Linked Data), centrándose en las conexiones entre conceptos y entidades [18].

Sin embargo, no todas las bases de conocimiento son grafos de conocimiento. Una característica clave de un grafo de conocimiento es que las descripciones de las entidades deben estar interrelacionadas entre sí. La definición de una entidad incluye a otra entidad. Así se forma el grafo. (Por ejemplo, A es B. B es C. C tiene D. A tiene D). Las bases de conocimiento sin estructura formal ni semántica, por ejemplo, la "base de conocimiento" de Q&A sobre un producto de software, tampoco representan un grafo de conocimiento. Es posible tener un sistema experto que tenga una colección de datos organizados en un formato que no sea un grafo, pero que utilice procesos deductivos automatizados, como un conjunto de reglas "si-entonces", para facilitar el análisis.

Existen ejemplos de grandes grafos de conocimiento como: Google Knowledge Graph[6], DBPedia [7], Geonames[8], Wordnet[9] o FactForge[10], que reunen millones de entidades y atributos enlazados en grandes estructuras enlazadas.

Razonador. Es un componente software que desempeña un papel crucial en el contexto de la Web Semántica, ya que los razonadores implementan la inferencia lógica y el procesamiento automatizado de información. Los razonadores están diseñados para analizar y comprender el significado de los datos semánticos, basándose en ontologías y reglas lógicas (SWRL) definidas en el dominio del problema. En esta Tesis se ha usado el razonador Pellet [19], que es capaz de realizar diversas tareas, como la clasificación y la deducción, lo que le permite extraer información implícita y establecer relaciones entre los conceptos y entidades presentes en el grafo de conocimiento. Mediante el razonamiento lógico y la inferencia basada en reglas, Pellet puede descubrir nuevos conocimientos a partir de los datos existentes, resolver inconsistencias y conflictos, además

[6]API de acceso a Google Knowledge Graph (última visita 13-10-2023) `https://developers.google.com/knowledge-graph?hl=es-419`
[7]Sitio web de DBPedia (última visita 13-10-2023) `https://es.dbpedia.org/`
[8]Sitio web de Geonames (última visita 13-10-2023) `https://www.geonames.org/`
[9]Sitio web de Wordnet (última visita 13-10-2023) `https://wordnet.princeton.edu/`
[10]Sitio web de FactForge (última visita 13-10-2023) `http://factforge.net/`

de ofrecer soluciones para mantener la coherencia de la información. Así, Pellet se convierte en una herramienta esencial que impulsa la precisión, el poder y el intercambio de conocimientos en los grafos de conocimiento.

Linked Data. En 2006 Tim Berners-Lee [20], el inventor de la "World Wide Web" publicó su visión personal de lo que deben cumplir los datos para ser considerados datos enlazados (en inglés "Linked Data"). En concreto definió cuatro reglas fundamentales, conocidas como "Las 4 Reglas del Linked Data". Estas reglas son:

- **Usar URIs como nombres de los recursos**: Los recursos (como entidades, conceptos o propiedades) deben tener identificadores únicos y persistentes conocidos como URIs (Uniform Resource Identifiers). Las URIs permiten identificar y acceder a los recursos en la web de manera unívoca.
- **Usar HTTP para resolver las URIs**: Las URIs deben utilizar el protocolo HTTP para permitir que los datos sean accesibles a través de la web. Esto facilita la recuperación y el intercambio de datos de manera estándar y global.
- **Proporcionar la información mediante los estandares RDF* o SPARQL**: Los recursos deben proporcionar información estructurada sobre sí mismos y sus relaciones utilizando el lenguaje de descripción de recursos RDF. Esto permite establecer enlaces y conexiones semánticas entre diferentes recursos, lo que enriquece y amplía el conocimiento asociado a los datos.
- **Incluir enlaces a otros recursos relacionados**: Los datos deben incluir enlaces a otros recursos relacionados, permitiendo así la navegación y el descubrimiento de información adicional. Estos enlaces ayudan a construir una red de datos interconectados, fomentando la exploración y la integración de conocimientos.

Posteriormente en 2010, Tim Berners-Lee, añadió un sistema para evaluar la apertura y vinculación de los datos. Se basa en la idea de que los datos pueden obtener "estrellas" a medida que cumplen con las reglas de los datos aboertos enalzados (en inglés "Linked Open Data"). El sistema de clasificación consta de cinco niveles, desde 1 hasta 5 estrellas, donde cada nivel representa un mayor grado de apertura y enlace de los datos. En la Tabla 2.2 se describen los cinco niveles.

Tabla 2.2: Sistema de estrellas propuesto por Tim Berners-Lee.

Nivel de estrellas	Descripción
⋆	Disponible en la web (en cualquier formato) pero con una licencia de uso abierta.
⋆ ⋆	Disponibles como datos estructurados legibles por una máquina.
⋆ ⋆ ⋆	Todo lo anterior, y además usar un formato no propietario.
⋆ ⋆ ⋆ ⋆	Todo lo anterior, pero usando estándares definidos por la W3C (RDF y SPARQL).
⋆ ⋆ ⋆ ⋆ ⋆	Todo lo anterior, añadiendo enlaces a otros datos.

En esta actualización, Tim Berners-Lee también declaró que estaba recibiendo presiones para añadir un requisito más a los datos publicados por las entidades gubernamentales y es que debería añadirse metadatos sobre los datos en sí, y que esos metadatos deberían estar disponibles en un catálogo, como por ejemplo CKAN [11].

[11] Página web de CKAN `https://ckan.org/`

Principios FAIR. Los principios FAIR (Localizable, Accesible, Interoperable, Reusable) [21] se aplican en el contexto de la Web Semántica y los grafos de conocimiento distribuidos para garantizar la efectiva gestión y utilización de los datos. En primer lugar, localizable se refiere a la capacidad de encontrar y acceder fácilmente a los datos. Esto implica el uso de identificadores persistentes y estándares de metadatos bien definidos que permitan su recuperación y descubrimiento. La accesibilidad se relaciona con la disponibilidad de los datos para su consulta y uso. Los datos deben ser accesibles de manera abierta y sin restricciones, preferiblemente utilizando estándares y protocolos comunes. La interoperabilidad es un principio clave que establece que los datos deben ser estructurados y representados de forma que puedan ser entendidos y utilizados por diferentes sistemas y aplicaciones. Esto implica el uso de ontologías y vocabularios compartidos, así como de formatos de datos estandarizados. Por último, la reutilización se refiere a la capacidad de los datos de ser utilizados en diferentes contextos y por diferentes usuarios. Esto implica la disponibilidad de licencias claras, la documentación adecuada y la adopción de prácticas que promuevan la reutilización de los datos.

Los autores desglosan los cuatros principios en los siguientes subelementos:

Localizable

- F1. Los metadatos se les asigna un identificador globalmente único y persistente.
- F2. Los datos se describen con metadatos ricos (definidos por R1 a continuación).
- F3. Los metadatos incluyen de manera clara y explícita el identificador de los datos que describen.
- F4. Los metadatos se registran o se indexan en un recurso de búsqueda.

Accesible

- A1. Los metadatos se pueden recuperar mediante su identificador utilizando un protocolo de comunicación estandarizado.
  - A1.1. El protocolo es abierto, gratuito y universalmente implementable.
  - A1.2. El protocolo permite un procedimiento de autenticación y autorización, cuando sea necesario.
- A2. Los metadatos son accesibles, incluso cuando los datos ya no están disponibles.

Interoperable

- I1. Los metadatos utilizan un lenguaje formal, accesible, compartido y ampliamente aplicable para la representación del conocimiento.
- I2. Los metadatos utilizan vocabularios que siguen los principios FAIR.
- I3. Los metadatos incluyen referencias a otros metadatos.

Reusable

- R1. Los metadatos se describen detalladamente con una pluralidad de atributos precisos y relevantes.
  - R1.1. Los metadatos se publican con una licencia clara y accesible para su uso.
  - R1.2. Los metadatos están asociados con una procedencia detallada.
  - R1.3. Los metadatos cumplen con los estándares comunitarios relevantes para el dominio.

2.2. Machine Learning

El aprendizaje automático, también conocido como *"Machine Learning"* (ML)[22] en inglés, es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las máquinas, clasificar, agrupar y predecir automáticamente a partir de datos. Estos algoritmos se basan en la idea de que las computadoras pueden analizar y reconocer patrones en grandes conjuntos de datos para tomar decisiones o hacer predicciones sin ser programadas explícitamente para un conjunto de datos concreto.

El origen del aprendizaje automático se remonta a mediados del siglo XX, cuando los investigadores comenzaron a imaginar formas de hacer que las máquinas pudieran aprender y adaptarse sin una programación manual. Estas redes estaban inspiradas en la estructura y el funcionamiento del cerebro humano, y se utilizaron para imaginar modelos capaces de reconocer patrones en datos complejos.

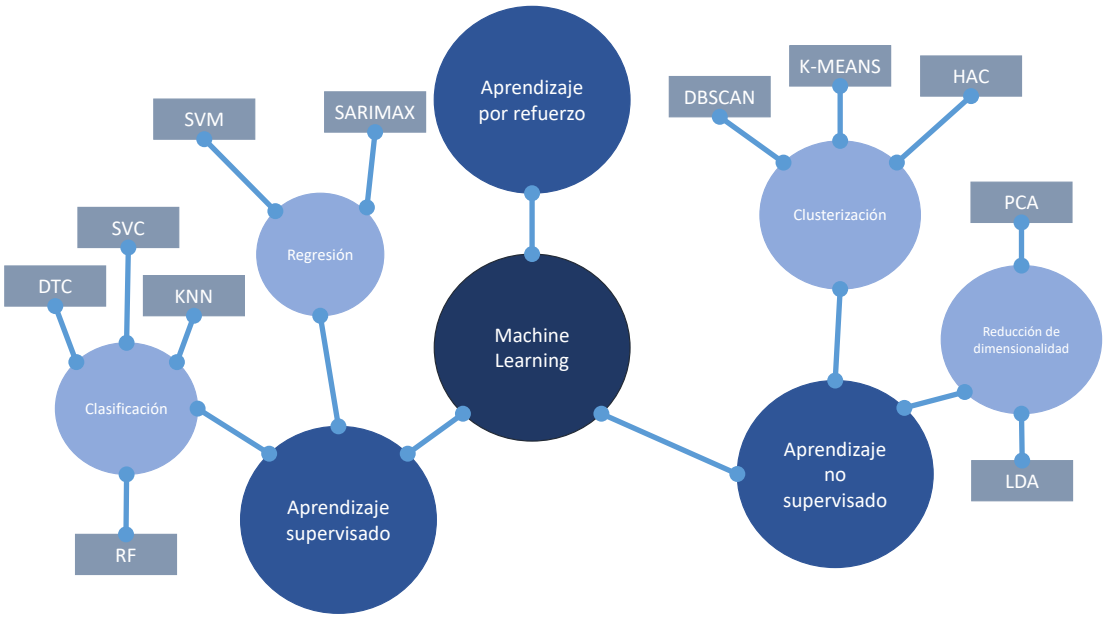

Figura 2.2: Algoritmos de Machine Learning

En los últimos años, el aprendizaje automático ha experimentado un crecimiento impresionante gracias a varios factores clave. En primer lugar, debido a la disrupción del *"Big Data"*, el ML se ha convertido en una de las áreas más prometedoras dentro del campo de la Inteligencia Artificial. A medida que la cantidad de datos disponibles crece exponencialmente, los algoritmos ML se han vuelto indispensables para extraer conocimiento, patrones y realizar predicciones significativas a partir de los datos. Estos algoritmos permiten a las máquinas aprender de los datos y mejorar su precisión a medida que se exponen a más información. Además, los avances en el hardware y en las técnicas de procesamiento, como el uso de unidades de procesamiento gráfico (GPU) y el cómputo en la nube, han acelerado el entrenamiento y la implementación de los algoritmos de aprendizaje automático. A continuación se describen los tipos de familias de algoritmos existentes actualmente.

- **Aprendizaje supervisado**. En este tipo de algoritmos a la máquina se le proporciona como entrada un conjunto de datos etiquetados; Es decir, se conoce la salida deseada. La máquina

aprende y va ajustando las correlaciones de las características de los datos de entrada con las etiquetas de salida. El objetivo del aprendizaje supervisado es que el modelo pueda generalizar correctamente a partir de los datos de entrenamiento y ser capaz de hacer predicciones precisas sobre nuevos datos de entrada no vistos anteriormente.

Para lograr esto, se utilizan diferentes algoritmos de aprendizaje supervisado, como regresión lineal, regresión logística, árboles de decisión, bosques aleatorios, máquinas de vectores de soporte, redes neuronales, SARIMAX, Prophet, entre otros. Cada algoritmo tiene sus propias características y suposiciones, pero todos comparten el objetivo común de aprender a partir de datos etiquetados para realizar predicciones.

- **Aprendizaje no supervisado**. En estos algoritmos, no se proporcionan etiquetas de salida a la máquina. En su lugar, el algoritmo busca patrones y estructuras ocultas en los datos sin ninguna guía externa. Este tipo de algoritmo se utiliza para definir agrupaciones, relaciones entre variables o anomalías en los datos.

  Algunos ejemplos comunes de técnicas de aprendizaje no supervisado incluyen el algoritmo de *"clustering k-means"*, el análisis de componentes principales (PCA), la agrupación jerárquica.

  Son algoritmos pensados para explorar y comprender conjuntos de datos complejos y puede ser utilizado en una amplia gama de aplicaciones, como el análisis de datos, la segmentación de grupos y la detección de anomalías.

- **Aprendizaje por refuerzo**. El aprendizaje por refuerzo se basa en la idea de aprender a través de probar y equivocarse. El agente inteligente explora diferentes acciones disponibles en su entorno y recibe retroalimentación en forma de recompensas o penalización, lo que le permite aprender qué acciones son las más beneficiosas. Estos algoritmos definen una función que permite establecer esa recompensa o castigo, para una decisión tomada así puede ajustar su comportamiento a medida que acumula experiencia.

  Este enfoque se ha aplicado con éxito en diversos campos, como juegos, robótica, optimización de recursos y toma de decisiones en tiempo real. El aprendizaje por refuerzo permite a los agentes inteligentes aprender de manera autónoma y adaptativa, lo que lo hace especialmente útil en situaciones donde no se dispone de un conjunto completo de datos etiquetados de antemano.

Las tecnologías de la Web Semántica juegan un papel crucial en la integración de datos y su organización de cara a ser empleados en los algoritmos ML, además de ofrecer un potencial enorme de cara a facilitar su explicabilidad e interpretabilidad. Las ontologías proporcionan un marco de definición formal basada en lógica de descripciones permitiendo una comprensión más profunda de cómo y por qué los algoritmos de machine learning llegan a ciertas conclusiones o realizan ciertas predicciones. Al utilizar ontologías y descripciones lógicas, la semántica puede ayudar a mapear y representar los procesos de inferencia y razonamiento utilizados por los algoritmos de machine learning. Esto permite una mayor transparencia y explicabilidad al revelar las reglas y relaciones que guían las decisiones del modelo.

2.3. e-Learning

La educación es una de las instituciones más importantes y antiguas de la humanidad, y su objetivo es difundir conocimientos y habilidades a la siguientes generaciones. Con el avance tecnológico y la globalización, la educación ha evolucionado de manera significativa, por lo que se ha vuelto más importante que nunca. Para lograr una educación de alta calidad, y debido al auge las tecnologías informáticas, es necesario implementar sistemas de *"e-Learning"* eficientes. Los sistemas

de *"e-Learning"* son herramientas tecnológicas diseñadas para apoyar el proceso educativo, desde la planificación hasta la evaluación y el seguimiento de los estudiantes. Estos sistemas permiten la gestión de la información y la planificación de las actividades, y ayudan a los docentes a realizar su trabajo de manera efectiva. Además, los sistemas de *"e-Learning"* también proporcionan a los estudiantes acceso a recursos y materiales educativos en línea, lo que puede mejorar su experiencia educativa y aumentar su capacidad de aprendizaje.

A continuación se describen los conjuntos de datos de *"e-Learning"* usados en el capítulo 3.

- **Moodle** [12] es una plataforma de gestión del aprendizaje en línea (*"Learning Management System"*, LMS) ampliamente utilizada en todo el mundo. Fue creada con el objetivo de proporcionar una herramienta accesible y fácil de usar para la gestión de cursos y recursos educativos en línea. Moodle es un software libre y de código abierto, lo que significa que es accesible y personalizable para adaptarse a las necesidades individuales de cada institución educativa. La plataforma está diseñada para ser intuitiva y fácil de usar tanto para los docentes como para los estudiantes, y cuenta con una amplia gama de funciones y herramientas que permiten a los docentes crear y gestionar cursos en línea, evaluar tareas, y comunicarse con los estudiantes. Moodle también permite a los estudiantes acceder a los materiales de curso y realizar tareas en línea, lo que les permite aprender a su propio ritmo y en su propio horario. Además, la plataforma también proporciona foros en línea para que los estudiantes y los docentes puedan discutir y colaborar.
- ***"Open University Learning Analytics Dataset"*** (OULAD) [23] es un conjunto de datos de aprendizaje abierto que fue recopilado por la Open University en el Reino Unido. Este conjunto de datos incluye información detallada sobre los estudiantes, su desempeño y su participación en el aprendizaje en línea.
- ***"COCO: Semantic-Enriched Collection of Online Courses at Scale with Experimental Use Cases"*** [24] es un conjunto de datos enfocado en la creación de una colección semánticamente enriquecida de cursos en línea a gran escala. El objetivo de este proyecto es mejorar la accesibilidad y la comprensión de la información sobre cursos en línea a través del uso de técnicas de enriquecimiento semántico, esta información incluye detalles sobre el contenido de los cursos, los objetivos de aprendizaje, los requisitos previos y opiniones de los estudiantes.

Desarrollar un modelo semántico de integración de datos en el dominio de e-Learning permite mejorar el análisis predictivo de sistemas de e-Learning. Al utilizar datos abiertos y enlazados, se pueden crear soluciones más genéricas y enriquecer los grafos de conocimiento con más individuos y propiedades con las que alimentar a los algoritmos de aprendizaje automático. La utilización de ontologías y tecnologías de integración de datos de e-Learning de diferentes fuentes y con diversos formatos proporciona una visión unificada, más completa y precisa de la información.

2.4. Open Banking

La gestión de transacciones financieras es un proceso crítico para las instituciones financieras, por lo que su eficiencia y seguridad son de suma importancia. En el entorno actual, las regulaciones financieras desempeñan un papel fundamental en la protección de los datos y la privacidad de los clientes. Una de las normativas más relevantes en Europa es la Directiva de Servicios de Pago 2 [13](PSD2, por sus siglas en inglés).

[12]Página web de Moodle `https://moodle.org/`
[13]Directiva (EU) 2015/2366 `http://data.europa.eu/eli/dir/2015/2366/oj`

La normativa PSD2 es una regulación que tiene como objetivo promover la competencia, la innovación y la seguridad en el ámbito de los servicios de pago. Entre sus disposiciones, destaca la apertura de los datos bancarios a terceros proveedores de servicios de pago, conocidos como TTPS (*"Third-Party Payment Service Providers"*), a través de interfaces de programación de aplicaciones (APIs).

En este contexto, los grafos de conocimiento emergen como una tecnología prometedora para la gestión de transacciones financieras en cumplimiento con la normativa PSD2. Como se ha definido anteriormente, un grafo de conocimiento es una estructura de datos que captura y organiza relaciones entre diferentes entidades, permitiendo una representación más completa y contextual del ecosistema financiero.

Mediante el uso de grafos de conocimiento, las instituciones financieras pueden mapear y visualizar las relaciones entre los diferentes actores, como clientes, cuentas bancarias, proveedores de servicios de pago y transacciones. Esto facilita la monitorización y la detección de patrones anómalos o fraudulentos en las transacciones financieras, mejorando la seguridad y la prevención del fraude.

Además, los grafos de conocimiento permiten una gestión más eficiente de los datos y la información, al proporcionar una visión holística de la red de transacciones financieras. Esto ayuda a las instituciones a obtener información valiosa sobre sus clientes, los riesgos y las oportunidades comerciales, lo que a su vez puede respaldar la toma de decisiones estratégicas y la identificación de nuevas oportunidades de negocio.

En este sentido, el informe *"Hype Cycle for Emerging Technologies in Finance"* [14] destaca los grafos de conocimiento como una de las tecnologías emergentes con un alto potencial de impacto en el sector financiero en el intervalo de 5 a 10 años. Por tanto, se refuerza y motiva la aplicacón de nuevos enfoques de la Web Semantica en este sector que se ha desarrollado en parte importante en esta Tesis Doctoral.

2.5. Seguridad en bases de datos orientadas a grafos

La gestión de la seguridad en el ámbito del Big Data se ha vuelto cada vez más compleja debido a la proliferación de amenazas cibernéticas y la expansión de las redes empresariales. En este contexto, el modelo de Control de Acceso Basado en Roles (RBAC, por sus siglas en inglés) ha sido ampliamente utilizado para garantizar la seguridad y el control de acceso en los sistemas de información.

Sin embargo, debido a este aumento de volumen de datos y la interconexión de diferentes fuentes de datos, el panorama de la ciberseguridad requiere una aproximación más avanzada y flexible. Es aquí donde el concepto de *"Cybersecurity Mesh Architecture"* entra en juego. Esta arquitectura propone un enfoque descentralizado y distribuido para la seguridad, donde los controles de seguridad se extienden más allá de las fronteras tradicionales de la organización y abarcan a los usuarios, dispositivos y aplicaciones en un entorno de malla conectada.

Para implementar de manera efectiva la *"Cybersecurity Mesh Architecture"* y abordar los desafíos asociados con la gestión de la seguridad, los grafos de conocimiento y las ontologías se vuelven herramientas clave. Un grafo de conocimiento distribuido es una representación estructurada de conocimiento interconectado, donde los conceptos y las relaciones se modelan como nodos y arcos respectivamente. Al ser distribuido, permite una mayor escalabilidad y flexibilidad en la gestión de la seguridad en entornos empresariales distribuidos y complejos.

Al combinar grafos de conocimiento distribuidos y ontologías con la gestión de la seguridad basada en RBAC, es posible lograr una mayor visibilidad y comprensión de los activos de información, los riesgos y las relaciones en un entorno empresarial. Los grafos de conocimiento distribuidos

[14] Sitio web del informe en Gartner (última visita 13-10-2023) `https://www.gartner.com/en/documents/4518699`

permiten la colaboración y la sincronización de información en tiempo real, facilitando la detección y respuesta ante amenazas de manera más efectiva. Al mismo tiempo, la ontología proporciona una capa de abstracción que entre diferentes sistemas y aplicaciones de seguridad. Esto es especialmente beneficioso en entornos donde existen diversas fuentes de datos y tecnologías de almacenamiento.

De manera similar al punto anterior, el informe Gartner *"Hype Cycle for Emerging Technologies"* destaca la relevancia de este enfoque reconociendo su potencial para mejorar la gestión de la seguridad y abordar los desafíos emergentes en el plazo superior a 10 años.

Parte II
Metodología, análisis y resultados

Capítulo 3

Modelo semántico de integración de datos para la mejora del análisis predictivo en plataformas de e-Learning

En los últimos años, los sistemas de gestión del aprendizaje (LMS) del inglés, *"Learning Management System"*, están adquiriendo una gran importancia en la educación *"online"*, ya que ofrecen plataformas flexibles para organizar una gran cantidad de recursos educativos, así como para establecer canales de comunicación eficaces entre profesores y alumnos. Estas plataformas atraen a un número cada vez mayor de usuarios que acceden continuamente, cargan y descargan recursos e interactúan entre sí durante sus procesos de enseñanza y aprendizaje. Esta tendencia se ha acelerado además con la irrupción de la COVID-19.

En este capítulo se presenta, el modelo semántico *e-LION* (*"e-Learning Integration ONtology"*) cuyo objetivo es operar como enfoque de integración de datos de diferentes bases de conocimiento en el dominio del aprendizaje electrónico o *"e-Learning"*. Con fines demostrativos, el modelo ontológico propuesto se puebla con fuentes de datos del mundo real, privadas y públicas, procedentes de diferentes sistemas de gestión de la docencia que hacen referencia a cursos universitarios de la titulación de Ingeniería del Software de la Universidad de Málaga y de la Open University. En este sentido, se trabaja con un conjunto de cuatro casos de estudio para su validación, que abarcan la consulta semántica avanzada de los datos para alimentar el modelado predictivo y la predicción de series temporales de las interacciones de los estudiantes en función de sus calificaciones finales, así como la generación de reglas de razonamiento SWRL para la clasificación del comportamiento de los estudiantes. Los resultados son prometedores y conducen al posible uso de e-LION como esquema mediador ontológico para la integración de nuevos modelos semánticos futuros en el dominio del *"e-Learning"*.

3.1. Introducción

En la última década, los avances en el acceso a las nuevas tecnologías experimentados por la mayor parte de la sociedad se reflejan también en la educación *"online"*. El acceso y uso de plataformas de gestión de la docencia se vio acelerado por la irrupción de la COVID-19, lo que llevó a las instituciones académicas a revisar sus estrategias educativas. En este contexto, una plétora de herramientas y recursos de aprendizaje en línea son usados, para facilitar una metodología similar al sistema tradicional, que conecta a profesores y alumnos de forma asíncrona para llevar a cabo un proceso de aprendizaje didáctico. Entre estas herramientas, los LMSs están adquiriendo una gran importancia en la educación *"online"*, ya que ofrecen plataformas en línea de integración flexible para organizar una gran cantidad de recursos educativos, así como para establecer canales de comunicación eficaces entre cualquier miembro de la comunidad educativa.

En consecuencia, los LMS atraen a un número cada vez mayor de usuarios que acceden continuamente, cargan y descargan recursos e interactúan entre sí durante sus procesos de enseñanza y aprendizaje. Esto conlleva la generación de grandes volúmenes de datos relacionados con el aprendizaje que pueden analizarse para apoyar a los profesores en la planificación de lecciones, cursos o titulaciones de grado o posgrado, así como a las administraciones en el plano estratégico universitario. Por ejemplo, es posible extraer cómo se relacionan las interacciones de los alumnos en el LMS con las calificaciones que obtienen, lo que de alguna manera permite a los profesores establecer una clasificación del rendimiento esperado del estudiantado en función de las interacciones en el sistema. Esto, sin duda, proporciona a los profesores acceso a nuevos conocimientos que, junto con sus experiencias, pueden conducirles a conocer con más precisión el comportamiento de los estudiantes.

Curiosamente, este contexto es adecuado, por ejemplo, para algoritmos de aprendizaje automático predictivo basados en datos de series temporales para predecir el número de visitas en el LMS. Estos algoritmos se han aplicado previamente en diferentes dominios [25, 26, 27].

Sin embargo, la gestión de tal cantidad de datos, normalmente procedentes de múltiples fuentes heterogéneas y con atributos que en ocasiones reflejan inconsistencias semánticas, constituye un reto emergente, por lo que se requieren esquemas comunes de definición e integración que permitan fusionarlos fácilmente, con el objetivo de alimentar eficientemente los modelos de aprendizaje automático. En este sentido, las tecnologías de web semántica surgen como un marco útil para la integración semántica de datos de aprendizaje eléctrónico provenientes de múltiples fuentes de información, permitiendo la consolidación, vinculación y consulta avanzada de forma sistemática.

El desarrollo de nuevas ontologías y su uso para la integración de datos está ampliamente documentado en la literatura existente en diferentes dominios de aplicación, tal y como se muestra en [28, 29, 30, 31, 32, 33]. Estas ontologías guían la creación de grafos de conocimiento que representan semánticamente los datos integrados y son la entrada de tareas analíticas, como se muestra en [28], o de razonamiento semántico, como también se desarrolla en [34, 35].

Las tecnologías de la web semántica en el ámbito del aprendizaje *"online"* son analizadas en la literatura actual en dos estudios recientes: [36] y [37], el primero orientado a los sistemas de recomendación en aprendizaje en línea impulsados por la semántica, el segundo identificando las tendencias actuales en ontologías de *"e-Learning"*. En este sentido, aún se identifican en estos trabajos un conjunto de cuestiones que requieren el abordaje de nuevas propuestas, principalmente relacionadas con la interoperabilidad, la vinculación, el enriquecimiento y el análisis de datos.

Con esta motivación, en este capítulo se propone el modelo semántico e-LION (*"e-Learning Integration ONtology"*) para operar como enfoque de consolidación de datos de diferentes bases de conocimiento de aprendizaje en línea, lo que conduce a enriquecer el análisis de datos. Este modelo consiste en una ontología OWL 2 (Ontology Web Language) que permite el desarrollo de mapeos semánticos de los datos, para transformar los datos originales en el formato estándar RDF,

creando un grafo de conocimiento. De este modo, los datos procedentes de fuentes heterogéneas se almacenan e integran en un repositorio RDF común, que ahora puede consultarse fácilmente. El objetivo principal es alimentar algoritmos de inteligencia artificial capaces de analizar patrones de interacción implícitos en los LMS realizados por una determinada comunidad de aprendizaje *"online"*.

Para validar el modelo semántico propuesto, se llevan a cabo una serie de funciones de mapeo y procesos de carga de volcados SQL para poblar e-LION con fuentes de datos privadas y públicas de diferentes LMS. En concreto, estas fuentes consisten en datos provenientes de la plataforma Moodle usada en la docencia de la titulación de Ingeniería del Software de la Universidad de Málaga, que se enriquecen con la integración del repositorio de datos Open University presentado en [23], así como la colección de datos de cursos online enriquecida semánticamente COCO propuesta por [24]. El enfoque semántico resultante permite la consulta avanzada de datos relativos a las interacciones de los estudiantes y sus rendimientos académicos para alimentar eficientemente modelos predictivos y visualizaciones. Además, gracias a la integración semántica, se llevan a cabo una serie de tareas de razonamiento para inducir nuevos conocimientos implícitos que permiten clasificar los diferentes comportamientos de los estudiantes.

Las principales contribuciones de este capítulo se exponen a continuación:

- Se propone el modelo semántico e-LION para operar como enfoque de consolidación de datos de diferentes bases de conocimiento de aprendizaje *"online"*, este modelo permite enriquecer el análisis de los algoritmos de aprendizaje automático. El modelo se codifica como una ontología OWL 2 que permite el desarrollo de mapeos semánticos al esquema de origen. La documentación de la ontología e-LION está disponible online [1].
- El poblado del modelo semántico se realiza con una serie de procesos de volcado de datos en lenguaje SQL de la plataforma Moodle del grado de Ingeniería del Software (Universidad de Málaga), junto con los datos de la Open University [23] y la colección de datos COCO [24]. En total, los datos contienen las interacciones de 43.228 asignaturas y 2.466.712 estudiantes a lo largo de varios años de funcionamiento. Estos datos se mapean al mismo grafo de conocimiento y se almacenan en el repositorio RDF, habilitando un punto SPARQL para su consulta.
- El enfoque semántico propuesto se valida mediante cuatro casos de estudio que comprenden el modelado predictivo y la predicción de series temporales de las interacciones de los estudiantes, así como el análisis de diferentes carácterísticas que permiten predecir las calificaciones finales; También se abarca la generación de reglas de razonamiento SWRL para la clasificación del comportamiento de los estudiantes.

Como se ha comentado anteriormente, la gestión de grandes volúmenes de datos almacenados por los sistemas de gestión de la docencia en línea es compleja debido a la procedencia heterogéna de las múltiples fuentes de datos. En este sentido, la definición de un esquema común en el dominio de la educación en línea constituye un reto emergente, el esquema se valida mendiante diferentes casos de uso. En estos casos de uso se muestra cómo se extrae la información del grafo de conocimiento y se usa para entrenar diversos algoritmos de aprendizaje automático.

3.2. Trabajos relacionados

Esta sección está dedicada a realizar una revisión de los trabajos relacionados en la literatura para situar nuestra propuesta dentro del estado actual del arte.

[1] Ontología e-LION disponible en la URL `http://ontologies.khaos.uma.es/e-lion`

Tabla 3.1: Resumen de las principales características de los trabajos relacionados en comparación con las ofrecidad por e-LION.

Ref.	Propósito	Audiencia Objetivo	Lenguaje	Aprendizaje Máquina	Raz.	Disp.
[38]	Recuperación de información	Estudiantes	OWL	PLN	No	No
[39]	Anotación de datos	Profesorado	OWL	No	No	No
[40]	Sistema de recomendación	Estudiantes	OWL	Minería de secuencias	No	No
[41]	Sistema de recomendación	Estudiantes	-	Agrupamiento difuso	No	No
[42]	Anotación de datos	Estudiantes/Profesorado	OWL	No	SWRL	No
[43]	Sistema de recomendación	Estudiantes	-	-	No	No
[44]	Anotación de datos	Profesorado	XML	-	No	No
[24]	Conjunto de datos	Estudiantes/Profesorado	JSON	KNN, PLN	No	Sí
[45]	Anotación de datos	Profesorado	OWL	No	No	No
[46]	Sistema de recomendación	Estudiantes	OWL	kMeans	No	No
e-LION	Integración de datos y Analítica	Estudiantes Profesorado	OWL 2	KNN, DT, SVM, RF, GNB, MLP, SARIMAX	SWRL	Sí

El uso de tecnologías de la web semántica en el dominio del aprendizaje *"online"* y en particular la conceptualización del conocimiento con ontologías en este ámbito, ha sido ampliamente estudiado en varias revisiones bibliográficas: [47, 48], a partir de las cuales han surgido una serie de propuestas recientes, abarcando los últimos cinco años. Más recientemente, en [49] se realiza una categorización de los estudios según el uso de ontologías en el contexto del aprendizaje y la educación, modelado y gestión del currículum, descripción del dominio de aprendizaje, descripción de datos del alumno y servicios de aprendizaje *"online"*.

Dentro de estas categorías, el uso de ontologías junto con técnicas de análisis de datos está ganando importancia, ya que se apoya en plataformas digitales de gestión de la docencia que permiten la generación de nuevas fuentes de datos sobre las actividades y comportamientos de los alumnos. Este aspecto ha sido ampliamente considerado en dos revisiones recientes: [50] y [36], aunque con especial atención en sistemas de recomendación basados en ontologías en ambos casos.

Por mencionar cronológicamente un conjunto representativo de contribuciones relacionadas con la propuesta, en 2016 se propone un sistema de recuperación de información en el campo del aprendizaje *"online"* basado en ontologías en [38], donde los autores analizaron la importancia de manejar conceptos basados en el procesamiento del lenguaje natural con herramientas como Wordnet o Hownet. También en 2016, los autores de [39] desarrollaron una plataforma de anotación semántica para evaluar las habilidades de los alumnos en una plataforma de *"e-Learning"* utilizando tecnologías de web semántica. Esta propuesta comprendía una ontología OWL anotada manualmente para la explotación de los datos de los alumnos con el fin de predecir su rendimiento en la formación. Últimamente, en 2017 se propone en [40] un método híbrido de recomendación basado en una ontología de recursos de los alumnos y la minería de patrones secuenciales para identificar los patrones históricos del alumno a partir de archivos de registro. Con un enfoque similar, en [41] se crea un sistema basado en el conocimiento bajo un esquema ontológico que permite el mapeo entre elementos para un recomendador de filtrado colaborativo. Se utiliza para personalizar la búsqueda del usuario en la web mediante un archivo que registra los clics del usuario. El registro de clics se utiliza a su vez para alimentar la base de conocimiento. También en esta línea, [42] propuso un ecosistema de aprendizaje *"online"* inteligente basado en un modelo ontológico con reglas de razonamiento SWRL. Este modelo consta de cuatro ontologías para recursos educativos, actividades de aprendizaje y métodos de enseñanza. El objetivo principal es proporcionar a los alumnos un entorno de aprendizaje personalizado.

Otro sistema de recomendación es el propuesto por [43], que se enriquece con métodos de aprendizaje automático para orientar a los estudiantes en la educación superior. Se trata de un enfoque basado en ontologías para anotar los requisitos, intereses, preferencias y capacidades de los estudiantes, con el objetivo de recomendar estudios superiores. También en este año, un modelo de enseñanza inteligente interdisciplinario se propone en [44] para mejorar la capacidad cognitiva de

los estudiantes, mientras que el apoyo a los profesores consiste en comprender el nivel de aprendizaje de los estudiantes. Como argumentan los autores, este modelo evalúa la capa de abstracción de la ontología de dominio y proporciona la base para mejorar el plan de enseñanza. De hecho, esta propuesta se valida a través de algunos casos de uso. En este sentido, [24] presentó COCO, una colección enriquecida semánticamente de cursos en línea que tiene como objetivo apoyar la experimentación y el diseño de servicios en el aprendizaje *"online"*. El conjunto de datos COCO incluye información recopilada de la plataforma de cursos online Udemy [2], permitiendo la generación de casos de uso orientados al análisis de datos de *"e-Learning"*.

Desde una perspectiva diferente, [45] propuso una metodología para construir una ontología basada en el esquema de la base de datos Moodle para el análisis de redes sociales. Esta propuesta modela la semántica de las influencias de las relaciones a partir de la topología del grafo de interacción del usuario. La ontología se construye mapeando directamente la estructura UML Moodle del Mount Orange School [3].

En [46] se presentó un marco ontológico utilizado para abordar el problema del arranque en frío en la recomendación de contenidos. En este modelo, la ontología propuesta está diseñada para cubrir el dominio contextual de los estudiantes y los recursos de aprendizaje. También incluye una agrupación multivariante de k-medias para evaluar la precisión del cálculo de la similitud de los estudiantes. Curiosamente, se midió la satisfacción del estudiantado con 40 participantes que utilizaron esta propuesta.

Recientemente, han aparecido en la literatura actual dos *"surveys"* completos que cubren diferentes aspectos en la intersección de la web semántica con el aprendizaje en línea. La revisión presentada por [36] está orientada a los sistemas de recomendación en el aprendizaje *"online"* impulsados por ontologías, que considera 28 artículos de revistas que combinan la semántica con la inteligencia artificial, las tecnologías informáticas, la educación, la psicología de la educación y las ciencias sociales. En segundo lugar, [37] discute una serie de tendencias actuales en ontologías de aprendizaje en línea, e identifica la interoperabilidad de datos como un tema clave que debe ser enfrentado en nuevos enfoques, no sólo en sistemas pertenecientes a las mismas instituciones, sino también en el contexto de diferentes fuentes, donde se requerirían ontologías para manejar similitudes y discrepancias. Esta cuestión específica es abordada por e-LION propuesto en este capítulo, junto con otras cuestiones diferentes.

En este sentido, la Tabla 3.1 contiene un resumen de las principales características de las propuestas seleccionadas en trabajos relacionados en comparación con las de e-LION. En concreto, se informa del propósito principal, la audicencia objetivo y el lenguaje usado para su diseño, además de las técnicas de aprendizaje automático utilizadas (cuando procede), y si han realizado razonamiento semántico y la disponibilidad de los recursos *"online"*.

Gran parte de estos enfoques están orientados a la generación de modelos semánticos, que utilizan ontologías para impulsar el desarrollo de sistemas de recomendación en diferentes aspectos del dominio de conocimiento del aprendizaje *"online"*. Sin embargo, hasta donde sabemos, ninguno de ellos está concebido para la tarea especial de integración de datos de múltiples fuentes en entornos de *"e-Learning"* para enriquecer los procesos analíticos de datos y las visualizaciones. El modelo semántico e-LION propuesto en este trabajo aspira a constituir un paso adelante en esa dirección.

3.3. Modelo semántico propuesto

Uno de los principales objetivos de la propuesta presentada en este capítulo es capturar, limpiar, consolidar e integrar datos de diferentes plataformas y repositorios de sistemas de gestión del

[2]Disponible en línea en la URL `https://www.udemy.com/`
[3]Disponible en línea en la URL `https://school.moodledemo.net/`

"e-Learning". Por este motivo, hemos optado por diseñar una aproximación semántica para compartir y unificar los datos implicados, a través de una ontología que modele el dominio en el que opera el sistema. En concreto, hemos definido una ontología OWL 2 para describir las principales características de las plataformas de gestión del aprendizaje en línea, tal y como se recomienda en [51],dónde se define el proceso de desarrollo de una ontología:

1. *Determinar el dominio y el alcance de la ontología.* Como punto de partida, para limitar el alcance de la ontología, se han seleccionado las variables que suelen almacenar la mayoría de los sistemas de gestión del aprendizaje *"online"*, por ejemplo: registro de interacciones, atributos de los alumnos, atributos de las tareas y de los envíos. Otro formalismo para describir componentes y datos interoperables del modelo podrían vincularse a partir de la ontología BIGOWL [52], dedicada a la anotación de flujos de trabajo analíticos de datos. Por simplicidad, se ha omitido para centrarse únicamente en el dominio de conocimiento de los sistemas de gestión del aprendizaje *"online"*.

2. *Considerar la reutilización de ontologías existentes.* Como se estudió en la Sección 3.2, no existen ontologías públicas que modelen completamente las interacciones de los usuarios y sus calificaciones en tareas. No obstante, se han considerado parcialmente dos ontologías relacionadas: en primer lugar, la ontología propuesta en [53] muestra un modelo básico de base de conocimientos de *"e-Learning"*, mientras que el enfoque mostrado en [54] tiene en cuenta las relaciones entre tareas y cursos. Estas ontologías no han sido reutilizadas directamente por e-LION, si bien, han servido de inspiración para el modelado de la ontología propuesta. Nuestra ontología pretende cubrir las necesidades de información relevantes para facilitar la minería y analítica de datos en el ámbito del *"e-Learning"*. Las ontologías existentes, como LOM (Learning Object Metadata)[4], CRSW (ReSIST Courseware Ontology)[5], Scorm y Tin Can API [6], se centran en un área específica del proceso de aprendizaje, es decir, los recursos de aprendizaje *"online"*, sin representar la diversidad de clases y métricas que almacenan los sistemas de gestión de la docencia *"online"*. Por otro lado, algunas ontologías de propósito general, como schema.org[7] y foaf[8], incluyen clases relacionadas con el aprendizaje electrónico. En la Sección 3.3.1 se describe cómo e-LION reutiliza algunas clases, principalmente las relacionadas con tipo de actividad, audiencia, y usuarios de plataformas de *"e-Learning"*.

3. *Enumerar términos importantes en la ontología.* Los términos importantes de la ontología se han extraído en una fase previa de análisis de requisitos. En esta fase, definimos el conjunto mínimo de variables que debían almacenarse. Algunos ejemplos de estos términos son: *tarea*, *entrega*, *usuario*, *curso*, *registro* y *matrícula*, entre otros.

4. *Definir clases y jerarquía de clases.* A partir de la lista de los términos más importantes, se han definido las clases de la ontología. La Figura 3.1 muestra el conjunto principal de clases de la jerarquía a partir de la clase superior *Thing*. Estas clases principales están relacionadas con otras clases para modelar las relaciones entre la información que contienen.

5. *Definir las propiedades de las clases.* Para relacionar las clases y definir atributos, se definen propiedades de objetos y datos en función del conjunto mínimo de variables. Ejemplos de propiedades de objeto son: un *Envío* pertenece a una *Tarea*, un *Envío* pertenece a un *Usuario*, un *Tarea* pertenece a un *Curso*, un *Usuario* está matriculado en un *Curso*, etc. Ejemplos de propiedades de tipos de datos son, el *Rol* de un usuario en un curso, la puntuación de un

[4]https://lov.linkeddata.es/dataset/lov/vocabs/lom
[5]https://lov.linkeddata.es/dataset/lov/vocabs/crsw
[6]https://xapi.com/
[7]https://schema.org/docs/developers.html
[8]http://xmlns.com/foaf/spec/

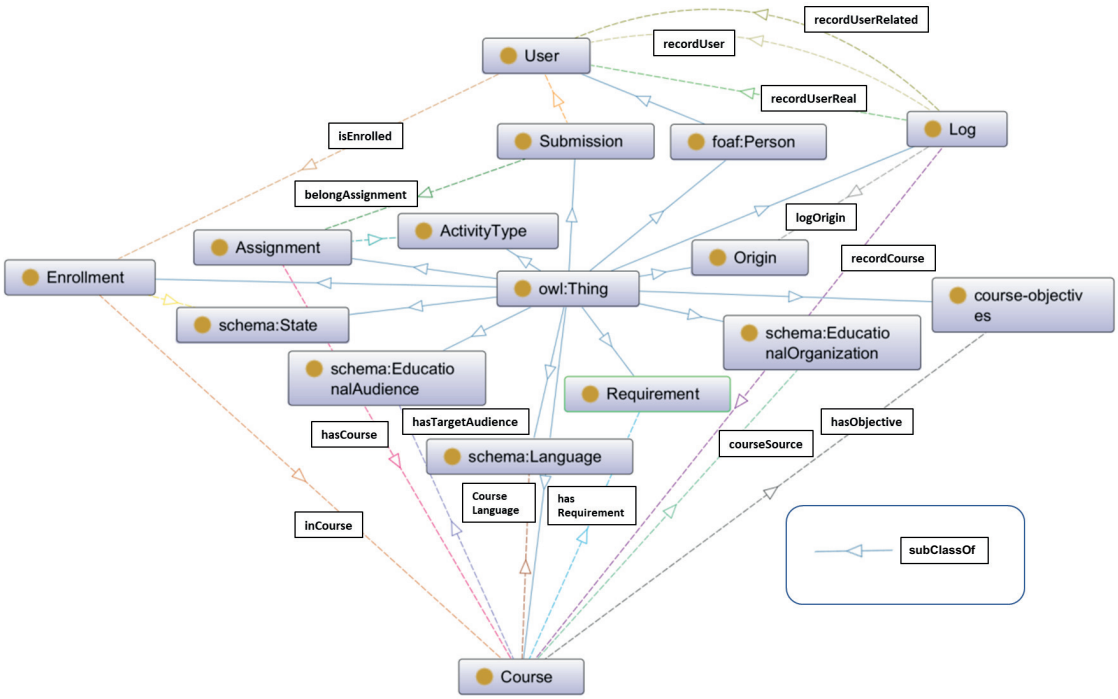

Figura 3.1: Visión general de la ontología e-LION. Las flechas continuas se refieren a subclases, mientras que las punteadas se refieren a propiedades.

envío, la marca de tiempo de una entrega, etc. Las tablas 3.2 - 3.7 describen un subconjunto representativo de propiedades de objetos y datos para una selección de las clases principales.

6. *Definir las restricciones de las propiedades.* Este paso incluye la definición de las restricciones de cardinalidad y valor. Las restricciones de valor se utilizan en e-LION para especificar los valores del tipo de datos en todas sus propiedades. Por ejemplo, el rango de la propiedad *logTimeCreated* está restringido a `dateTime`, mientras que en el caso de la propiedad *assignmentWeight* el rango está restringido a `int`.

7. *Crear instancias.* Las instancias (individuos en OWL) corresponden a los datos específicos obtenidos de las interacciones de los alumnos a partir del sistema gestor de aprendizaje *"online"*. Los individuos se obtendrán mapeando los datos de Moodle (volcados SQL) y otros LMMs a RDF según el esquema definido en la ontología e-LION.

3.3.1. Modelo ontológico

Tras aplicar la metodología anterior, se ha desarrollado la ontología e-LION que comprende un total de 15 clases (grupos de individuos con los mismos atributos), 17 propiedades de objeto (representan relaciones entre individuos), 78 propiedades de datos (atributos de los individuos) y 193 axiomas de restricción.

Por simplicidad, describimos aquí un subconjunto representativo de las principales propiedades de objeto y de datos de las clases que componen la ontología, que se formalizan mediante la sintaxis de la lógica de descripción OWL-DL (definida en la Tabla 2.1). Estas clases son: *Assignment*,

Tabla 3.2: Clase *"Course"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica Descriptiva
courseSource	$\exists$ courseSource Thing $\sqsubseteq$ Course $\top \sqsubseteq \forall$ CourseSource schema:EducationalOrganization
Propiedades de datos	**Lógica Descriptiva**
courseId	$\exists$ courseId Datatype Literal $\sqsubseteq$ Course $\top \sqsubseteq \forall$ courseId Datatype string
coursePresentationLength	$\exists$ coursePresentationLength Datatype Literal $\sqsubseteq$ Course $\top \sqsubseteq \forall$ coursePresentationLength Datatype int
courseUrl	$\exists$ CourseUrl Datatype Literal $\sqsubseteq$ Course $\top \sqsubseteq \forall$ CourseUrl Datatype string
courseDescription	$\exists$ courseDescription Datatype Literal $\sqsubseteq$ Course $\top \sqsubseteq \forall$ courseDescription Datatype string
courseClicksAVG	$\exists$ courseClicksAVG Datatype Literal $\sqsubseteq$ Course $\top \sqsubseteq \forall$ courseClicksAVG Datatype int

Tabla 3.3: Clase *"User"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica Descriptiva
isEnrolled	$\exists$ isEnrolled Thing $\sqsubseteq$ User $\top \sqsubseteq \forall$ isEnrolled Enrollment
Propiedades de datos	**Lógica Descriptiva**
userId	$\exists$ userId Datatype Literal $\sqsubseteq$ User $\top \sqsubseteq \forall$ userId Datatype int
userBiography	$\exists$ userBiography Datatype Literal $\sqsubseteq$ User $\top \sqsubseteq \forall$ userBiography Datatype string
userProfileUrl	$\exists$ userProfileUrl Datatype Literal $\sqsubseteq$ User $\top \sqsubseteq \forall$ userProfileUrl Datatype string
userJobTitle	$\exists$ userJobTitle Datatype Literal $\sqsubseteq$ User $\top \sqsubseteq \forall$ userJobTitle Datatype string

Course, *Enrollment*, *Log*, *Submission*, *Origin* and *User*. Además, las clases *xapi: ActivityType, courseware: course-objectives, foaf: Person, schema: EducationalAudience, schema: EducationalOrganization, schema: Language, schema: State* y *Requirement* se han incluido para tener en cuenta también otros repositorios semánticos existentes, como la colección COCO, permitiendo enlazar los datos. Cada clase requiere un conjunto de propiedades para ser modelada, es decir, un individuo que satisface esas propiedades se considera miembro de esa clase. La ontología completa se desarrolla en el archivo OWL `"e-LION.owl"`, disponible en el enlace [9]. A continuación, se ofrece una descripción de las principales clases de e-LION:

- ***Course***. Esta clase representa el conjunto de cursos que están registrados en el sistema de gestión del aprendizaje *"online"*. Define tres propiedades principales (entre otras) como se describe en la Tabla 3.2: *courseId*, para identificar unívocamente cada curso; *coursePresentationLength*, que representa la duración del curso en días; y *courseSource*, que registra la fuente de datos asociada al curso.
- ***User***. Define el conjunto de usuarios registrados en el sistema de gestión del aprendizaje *"online"*. Esta clase tiene una propiedad de datos *userId* para identificar a cada usuario, independientemente de que sea profesor o alumno, así como una propiedad de objeto *isEnrolled*

[9] Ontología e-LION `https://github.com/KhaosResearch/e-lion`

Tabla 3.4: Clase *"Assignment"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica Descriptiva
hasCourse	∃ hasCourse Thing ⊑ Assignment ⊤ ⊑ ∀ hasCourse Course
Propiedades de datos	**Lógica Descriptiva**
assignmentAllowSubmissionsFromDate	∃ assignmentAllowSubmissionsFromDate Datatype Literal ⊑ Assignment ⊤ ⊑ ∀ assignmentAllowSubmissionsFromDate Datatype dateTime
assignmentDueDate	∃ assignmentDueDate Datatype Literal ⊑ Assignment ⊤ ⊑ ∀ assignmentDueDate Datatype dateTime
assignmentName	∃ assignmentName Datatype Literal ⊑ Assignment ⊤ ⊑ ∀ assignmentName Datatype string
assignmentTimeModified	∃ assignmentTimeModified Datatype Literal ⊑ Assignment ⊤ ⊑ ∀ assignmentTimeModified Datatype dateTime
assignmentWeight	∃ assignmentWeight Datatype Literal ⊑ Assignment ⊤ ⊑ ∀ assignmentWeight Datatype int

que representa que un usuario forma parte de una asignatura. La Tabla 3.3 contiene la lógica de descripción de estas propiedades.

- ***Assignment***. Es una clase importante ya que representa las tareas propuestas por los profesores y entregadas por los alumnos. La propiedad del objeto *hasCourse* conecta una tarea con un curso determinado. La clase *Assignment* define un total de 29 propiedades de datos para modelar la configuración de la tarea, tales como: *assignmentDueData*, para considerar la fecha de entrega de una tarea; *assignmentName*, para recoger el nombre y la descripción de una tarea; *assignmentMaxAttempts*, para establecer el número máximo de intentos permitidos; y *assignmentWeight*, para definir la ponderación de la tarea en el curso. La Tabla 3.4 contiene un resumen de las propiedades de la clase *Assignment*, por lo que la lista completa puede extraerse del archivo de ontología en formato OWL.

Tabla 3.5: Clase *"Submission"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica Descriptiva
belongAssignment	∃ belongAssignment Thing ⊑ Submission ⊤ ⊑ ∀ belongAssignment Assignment
belongsUser	∃ belongsUser Thing ⊑ Submission ⊤ ⊑ ∀ belongsUser User
Propiedades de datos	**Lógica Descriptiva**
submissionId	∃ submissionId Datatype Literal ⊑ Submission ⊤ ⊑ ∀ submissionid Datatype string
submissionScore	∃ submissionScore Datatype Literal ⊑ Submission ⊤ ⊑ ∀ submissionScore Datatype float
submissionTimeCreated	∃ submissionTimeCreated DatatypeLiteral ⊑ Submission ⊤ ⊑ ∀ submissionTimeCreated Datatype dateTime
submissionTimeModified	∃ submissionTimeModified Datatype Literal ⊑ Submission ⊤ ⊑ ∀ submissionTimeModified Datatype dateTime

- ***Submission***. Esta clase conecta las anteriores, ya que representa las entregas realizados por un usuario respecto a una determinada tarea planteada en un curso. La clase *Entrega*

Tabla 3.6: Clase *"Enrollment"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica Descriptiva
inCourse	∃ inCourse Thing ⊑ Enrollment ⊤ ⊑ ∀ inCourse Course
Propiedades de datos	**Lógica Descriptiva**
enrollmentAgeBand	∃ enrollmentAgeBand Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentAgeBand Datatype string
enrollmentDateRegistration	∃ enrollmentDateRegistration Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentDateRegistration Datatype dateTime
enrollmentRole	∃ enrollmentRole Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentRole Datatype string
enrollmentRating	∃ enrollmentRating Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentRating Datatype float
enrollmentRatingDate	∃ enrollmentRatingDate Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentRatingDate Datatype dateTime
typeOfUser	∃ typeOfUser Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ typeOfUser DataRange {"Looker"Datatype string , "Passive" Datatype string, "Active" Datatype string}
enrollmentNumberOfClicks	∃ enrollmentNumberOfClicks Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentNumberOfClicks Datatype int
enrollmentNumberOfSubmissions	∃ enrollmentNumberOfSubmissions Datatype Literal ⊑ Enrollment ⊤ ⊑ ∀ enrollmentNumberOfSubmissions Datatype int

registra en gran medida las interacciones y actividades realizadas por los usuarios, por lo que también es una clase interesante ya que considera información sobre las interacciones de los usuarios. La Tabla 3.5 contiene la lógica de descripción de las dos propiedades de objeto definidas para esta clase: *belongAssignment*, que indica la tarea en la que se realiza la entrega, y *belongsUser* que hace referencia al usuario que realiza el envío de la tarea. Además, esta clase considera un conjunto de propiedades de datos (también descritas en la Tabla 3.5) para cubrir la información sobre un envío, tales como: *submissionAttempNumber* para indicar el número de intentos de entrega, *submissionId* es el identificador asignado al envío por la plataforma, *submissionLatest* indica si se trata del último intento de envío realizado por ese usuario en esa tarea, *submissionStatus* para indicar el estado del envío (borrador o enviado), *submissionTimeCreated* para recoger la marca de tiempo del primer envío en esta tarea, y *submissionTimeModified* para indicar la marca de tiempo en caso de que se modifique el envío.

- ***Enrollment*** representa la inscripción de alumnos en cursos. Para ello, la propiedad de objeto *inCourse* especifica el curso en el que se realiza la matrícula del usuario. Además, esta clase se define con un conjunto de propiedades de datos como se muestra en la Tabla 3.6 con sus descripciones lógicas. Algunas de las propiedades más interesantes son: *enrollmentRole*, que indica el rol del usuario en el curso (alumno, profesor, administrador, etc.), *enrollmentFinalResult* para establecer la nota final del alumno en la asignatura y *enrollmentDateRegistration*, esta última indicando la fecha de matriculación en la asignatura.

- ***Log*** también es una clase importante ya que cubre los eventos realizados por los usuarios en la plataforma de *"e-Learning"*. Cuenta con un conjunto de propiedades de objeto y propiedades de datos que se describen en la Tabla 3.7. Entre estas propiedades, un subconjunto de ellas que cabe mencionar serían: *logEduLevel* que representa a qué tipo de usuario pertenece el

Tabla 3.7: Clase *"Log"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica Descriptiva
recordCourse	$\exists$ recordCourse Thing $\sqsubseteq$ Log $\top \sqsubseteq \forall$ recordCourse Course
recordUser	$\exists$ recordUser Thing $\sqsubseteq$ Log $\top \sqsubseteq \forall$ recordUser User
recordUserReal	$\exists$ recordUserReal Thing $\sqsubseteq$ Log $\top \sqsubseteq \forall$ recordUserReal User
recordUserRelated	$\exists$ recordUserRelated Thing $\sqsubseteq$ Log $\top \sqsubseteq \forall$ recordUserRelated User
logOrigin	$\exists$ logOrigin Thing $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logOrigin Origin
Propiedades de datos	**Lógica Descriptiva**
logAction	$\exists$ logAction Datatype Literal $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logAction Datatype string
logEduLevel	$\exists$ logEduLevel Datatype Literal $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logEduLevel Datatype int
logId	$\exists$ logId Datatype Literal $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logId Datatype string
logSumClick	$\exists$ logSumClick Datatype Literal $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logSumClick Datatype int
logTarget	$\exists$ logTarget Datatype Literal $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logTarget Datatype string
logTimeCreated	$\exists$ logTimeCreated Datatype Literal $\sqsubseteq$ Log $\top \sqsubseteq \forall$ logTimeCreated Datatype dateTime

evento, por ejemplo, si el evento fue generado por un profesor entonces el campo contiene el valor 1, mientras que si fue generado por un estudiante, contiene el valor 2; *logAction* describe el tipo de acción que ha realizado el usuario (los valores más comunes para esta propiedad son "view" y "submitted"). Algunos sistemas registran los datos de forma agregada, por lo que esta información se almacena en *logSumlick* y *logTimeCreated* para indicar la marca de tiempo en la que se registró el evento.

3.3.2. Consolidación de datos

Una vez diseñado el modelo ontológico, se lleva a cabo una estrategia de consolidación de datos que permita la integración de las diferentes fuentes de datos, de acuerdo con el modelo definido. La Figura 3.2 muestra una visión general de esta estrategia, donde la caja terminológica (TBox) define el vocabulario con conceptos y relaciones en el dominio del e-Learning. Dentro de esta TBox, e-LION se define usando OWL 2 según el cual, los conceptos y las relaciones se representan mediante clases y propiedades de datos o propiedades de objetos, respectivamente. Esta ontología permite el enlace con otras ontologías educativas orientadas a diferentes aspectos, tales como: recomendación, currículo, material didáctico, MOOCs, etc. [24] [55], así como la alineación con otros datos enlazados externos[10] en diferentes dominios (DBPedia[11], Geonames[12], FOAF[13], etc.).

En un nivel diferente, los axiomas asertivos (ABox) considera todas las instancias del dominio de conocimiento que implican los datos relacionados con el LMS. Estas instancias se almacenan en formato tripletas RDF en un repositorio implementado por Stardog[14] con capacidades de persistencia y razonamiento. Para ello, se han implementado una serie de funciones de mapeo para convertir los datos procedentes de las distintas fuentes a RDF, siguiendo el esquema definido en e-LION.

Por tanto, en el caso de los datos de Moodle, éstos se mapean a partir de un conjunto de volcados SQL de una base de datos relacional relativa a la titulación de Ingeniería del Software de la Universidad de Málaga. Estos datos se utilizan por primera vez en este estudio y contienen la información anonimizada de las interacciones realizadas por los usuarios en esta plataforma LMS

[10] Open Linked Data Cloud `https://lod-cloud.net/`
[11] `https://wiki.dbpedia.org/`
[12] `https://www.geonames.org/`
[13] `http://www.foaf-project.org/`
[14] `http://www.stardog.com/`

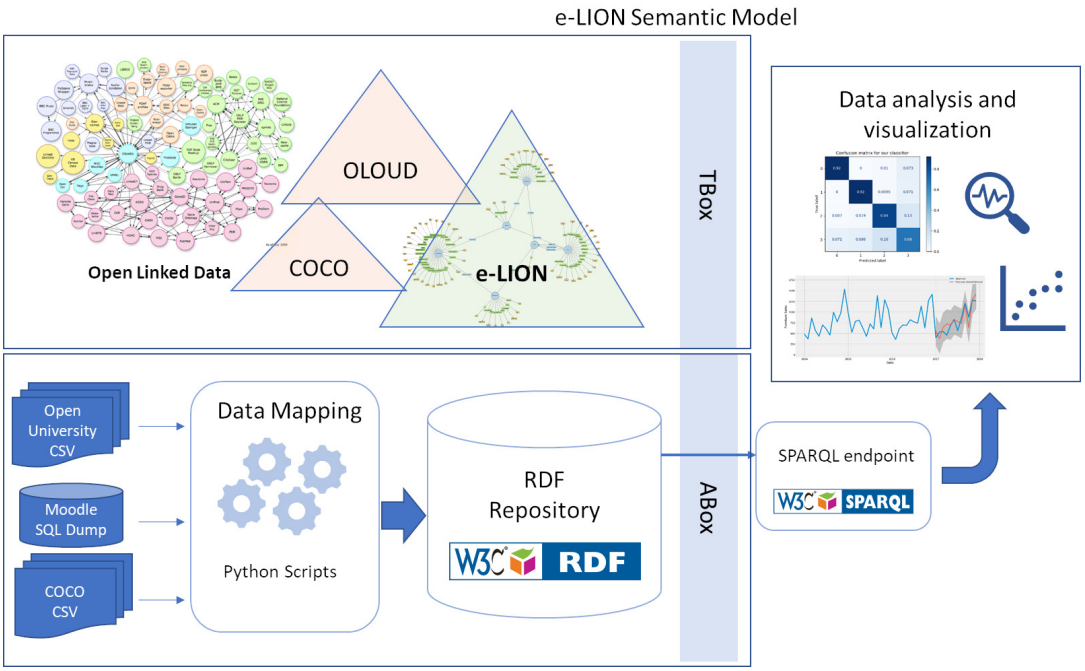

Figura 3.2: Visión general del modelo semántico e-LION.

(Moodle). Este conjunto de datos contiene información de 8.524 estudiantes en 93 cursos, 1.235.063 registros de interacciones, 1.342 tareas y 28.270 entregas. Una segunda fuente de datos se integra a partir del LMS de la Open University (OULAD), que se publicó para apoyar la investigación de minería de datos educativos [23]. Este conjunto de datos contiene datos de las interacciones de 32.593 estudiantes en 22 cursos, 10.655.280 registros de interacciones, 173.913 entregas y 206 tareas. También considera información demográfica, así como registros de interacción de los estudiantes con los materiales y calificaciones, tanto de las tareas como de la nota final de la asignatura. El conjunto de datos OULAD se proporciona en archivos tabulares CSV, por lo que las funciones de mapeo están adaptadas a este tipo de formato. Del mismo modo, el conjunto de datos COCO [24] también se proporciona en formato CSV, aunque consta de diferentes atributos que también han sido integrados en la ontología e-LION. Como se ha comentado anteriormente, el dataset COCO incluye información recogida de la plataforma Udemy de cursos *"online"*, permitiendo la generación de casos de uso orientados al análisis de datos de entornos de *"e-Learning"*. Esta tercera fuente de datos comprende 43.113 cursos en categorías de dos niveles e idiomas, conteniendo cada curso una media de 43 lecciones. También incluye un número de 2.436.677 estudiantes que interactuaron con 4.584.313 valoraciones y 2.453.800 comentarios.

La razón principal de integrar estas tres fuentes de datos es constituir una primera prueba de concepto para la validación del modelo semántico e-LION propuesto, ya que constituyen plataformas de aprendizaje *"online"* heterogéneas, que comprenden datos privados de Moodle, datos públicos de la Open University y el conjunto de datos académicos COCO de los cursos *"online"* de Udemy. En segundo lugar, el repositorio de datos resultante puede ampliarse con otros conjuntos de datos diferentes relacionados con el aprendizaje *"online"*, mediante la anotación semántica y la asignación de sus atributos de acuerdo con la estructura de la ontología e-LION.

Llegados a este punto, al tener los datos consolidados en el repositorio RDF común, ya es posible consultarlos desde un punto de consulta SPARQL, independientemente de la fuente de los datos,

su estructura o la sintaxis del formato original. De esta forma, los algoritmos de machine learning utilizados para realizar análisis exploratorios y predictivos en los casos de uso son alimentados con la información requerida relativa a las interacciones de los alumnos, las visualizaciones de los usuarios, el número de entregas, las calificaciones, de forma que los datos resultantes pueden ser agrupados por fecha, asignatura, etc., con consultas avanzadas en SPARQL. Un ejemplo de ello puede observarse en la Consulta 3.1, que se ejecuta para unificar los accesos a los datos correspondientes al número total de entregas realizadas por cada alumno en una asignatura. La Tabla 3.8 muestra parcialmente los resultados obtenidos a partir de esta consulta.

Simples consultas similares permiten obtener otro tipo de información, como el número de visitas realizadas por los alumnos en distintos periodos. En este sentido, la Figura 3.3 muestra una serie temporal de las visitas acumuladas por los alumnos en semanas. En este gráfico se puede observar que existen patrones temporales en las visitas, disminuyendo en periodos vacacionales (Navidad y verano), mientras que aumentan con ciertos picos en febrero, relacionados con las fechas de evaluación final de las asignaturas.

Por lo tanto, ahora es posible realizar un seguimiento de las interacciones de los alumnos en la plataforma de datos integrados, así como su posible correlación con el desarrollo de los alumnos en las asignaturas y las calificaciones finales obtenidas. También puede ser útil para comprender qué estrategias funcionan, así como para detectar cuándo un alumno se está desviando del seguimiento de la asignatura.

Fragmento de código 3.1: Ejemplo de consulta del número de envíos por usuario y curso.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (COUNT(?submission) AS ?count_sub) ?userid ?courseid
WHERE{
    ?course elion:courseId ?courseid.
    ?user elion:userId ?userid.
    ?submission elion:belongsUser ?user.
    ?submission elion:belongAssignment ?assigment.
    ?assigment elion:hasCourse ?course.
}GROUP BY ?userid ?courseid
```

Tabla 3.8: Ejemplos de resultados obtenidos por la Consulta 3.1.

userid	courseid	count_sub
629507	BBB2014B	10
629081	CCC2014B	3
2689210	FFF2014B	11
9485	73	2
11273	68	2

Una ventaja adicional de utilizar el enfoque semántico propuesto es la posibilidad de conectar el repositorio RDF con otros datos enlazados externos. Esto requiere una adaptación mínima para establecer qué clases y qué propiedades tienen un significado semántico equivalente, como se hecho con OLOUD y COCO.

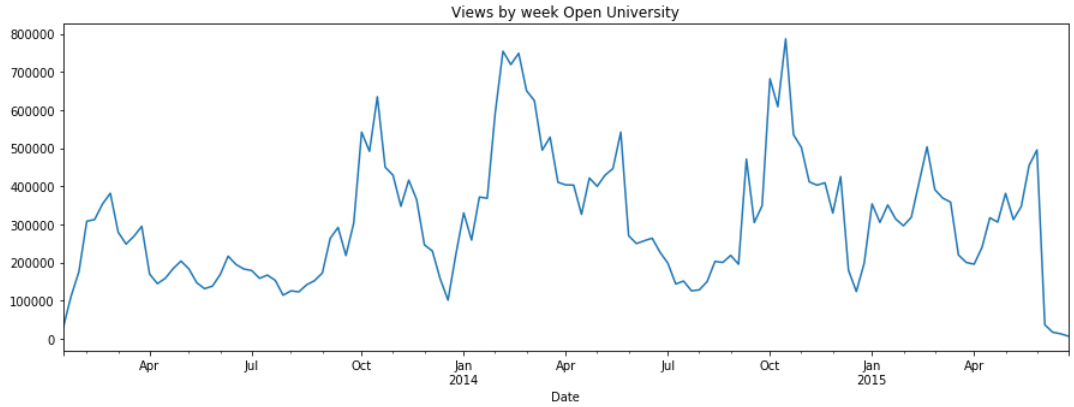

Figura 3.3: Gráfico de la serie temporal de las visualizaciones realizadas por los estudiantes de la Open University agrupadas por semanas en un periodo de dos años.

3.4. Validación

El modelo semántico propuesto proporciona un amplio conjunto de atributos de los estudiantes, asignaturas, entregas de trabajos e interacciones de los usuarios que tienen lugar en el sistema de gestión del aprendizaje *"online"*, junto con las calificaciones obtenidas en las asignaturas, permitiendo un análisis avanzado de los datos.

En términos de validación, en esta sección se llevan a cabo una serie de casos de estudio, consistentes en: predicción de la nota del alumno en la evaluación continua, predicción de la nota final del alumno, predicción de las series temporales de visitas de los alumnos y análisis de razonamiento para la clasificación del comportamiento de los alumnos. Estos casos de estudio se han elaborado para cubrir aspectos importantes de la propuesta, como por ejemplo: permitir una serie de consultas SPARQL a partir de datos integrados de fuentes diferentes, que se utilizan como conjuntos de entrenamiento y prueba para alimentar algoritmos de aprendizaje automático en tareas de modelado predictivo, así como permitir el razonamiento semántico basado en reglas para ilustrar cómo inferir nuevo conocimiento.

3.4.1. Caso práctico I: Predicción de la nota del alumno en evaluación continua

La predicción de calificaciones es una de las principales tareas de análisis de datos en educación, ya que permite a los profesores planificar y supervisar sus cursos con antelación, así como adoptar actividades de corrección en caso de desviaciones. En este primer caso de estudio, se entrenan una serie de modelos de clasificación supervisada para predecir las calificaciones de los alumnos en evaluación continua. Las principales características que alimentan los modelos son las visitas realizadas por los alumnos, las entregas realizadas y la diferencia en días entre el envío y las fecha de entrega. Una vez generados los modelos de predicción con información de cursos anteriores, pueden ser utilizados para predecir la calificación de aquellos cursos de la plataforma Moodle, que aún se encuentran sin calificación.

Siguiendo la tendencia actual de la educación *"online"*, que de hecho se ha visto incrementada por la situación mundial de pandemia de COVID-19, las clases objetivo se discretizan en aprobado/suspenso. Esto se centra en un sistema de calificación binario, lo que significa que no se registrará ninguna calificación con letras, sino que los estudiantes sólo obtendrán éxito en función

de si han realizado un trabajo satisfactorio en la clase. Por lo tanto, las calificaciones de los alumnos se han discretizado en 2 clases (Aprobado y Suspenso), según los rasgos que los caracterizan, como el número de entregas, el número de vistas y la diferencia de días desde la fecha de entrega de las tareas, es decir, la etiqueta "Aprobado" se refiere a los alumnos con un alto índice de entregas y un alto índice de vistas, mientras que la etiqueta "Suspenso" significa un bajo índice de entregas en las tareas y un bajo nivel de vistas.

Fragmento de código 3.2: Ontiene el número de visualizaciones por usuario y curso.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (SUM(?numclick) AS ?sum_clicks) ?userid ?courseid
WHERE{
  ?x elion:logEduLevel ?edulevel.FILTER (?edulevel = 2)
  ?x elion:recordUser ?user.
  ?x elion:recordCourse ?course.
  ?x elion:logSumClick ?numclick.
  ?course elion:courseSource elion:openUniversity.
  ?course elion:courseId ?courseid.
  ?user elion:userId ?userid
}GROUP BY ?userid ?courseid
```

En concreto, para este análisis se seleccionan una serie de atributos del repositorio RDF: *sum_click*, *id_student*, *code_module*, *code_presentation*, *weight*, *date_submitted*, *id_assessment* y *score*; que se obtienen a partir de consultas SPARQL que comprenden diferentes propiedades de clase de la ontología e-LION. En este sentido, se utiliza la Consulta 3.2 para calcular el número de vistas por usuario y curso. Esta consulta selecciona las tripletas cuyo nivel educativo es igual a 2, ya que este nivel corresponde a las interacciones de los estudiantes. Además, filtra el origen de los datos a los de la Open University y finalmente agrupa los resultados aplicando la suma al número de clics de un usuario en un curso.

Fragmento de código 3.3: Obtiene la calificación ponderada por usuario y curso.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (SUM(?w/100*?score/10) AS ?weight_score) ?userid ?courseid
WHERE{
  ?course elion:courseSource elion:openUniversity.
  ?course elion:courseId ?courseid.
  ?user elion:userId ?userid.
  ?assigment elion:hasCourse ?course.
  ?assigment elion:assignmentWeight ?w. FILTER(?w < 100)
  ?submission elion:belongAssignment ?assigment.
  ?submission elion:belongsUser ?user.
  ?submission elion:submissionScore ?score.
}GROUP BY ?userid ?courseid
```

La Consulta 3.3 devuelve los datos que serán discretizados para posteriormente ser utilizados como etiqueta (Aprobado/Suspenso). Esta consulta selecciona las tripletas cuyo origen es la Open University, la puntuación obtenida en la entrega y el peso que tiene en la calificación continua. Además, se filtran aquellos pesos superiores o igules a 100, ya que estas tareas corresponden a exámenes, que no pertenecen a la evaluación continua. Finalmente, la consulta se agrupa por usuario y asignatura aplicando la suma de las calificaciones teniendo en cuenta sus ponderaciones.

Del mismo modo, el número de entregas se puede calcular con la Consulta SPARQL 3.4. Se filtran las tripletas de la fuente de datos Open University y se seleccionan las entregas realizadas

por los usuarios, para posteriormente agrupar los resultados aplicando el recuento de envíos de un usuario en una asignatura.

Fragmento de código 3.4: Obtiene el número de entregas por usuario y curso.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (COUNT(?submission) AS ?count_sub) ?userid ?courseid
WHERE{
  ?course elion:courseSource elion:openUniversity.
  ?course elion:courseId ?courseid.
  ?user elion:userId ?userid.
  ?submission elion:belongsUser ?user.
  ?submission elion:belongAssignment ?assigment.
  ?assigment elion:hasCourse ?course.
}GROUP BY ?userid ?courseid
```

Fragmento de código 3.5: Obtiene la diferencia de días por usuario y curso.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (DAY(?date_diff_with_hours) AS ?diff_days)
       ?userid ?courseid
WHERE{
  SELECT (SUM(?fdate - ?timecreated) AS ?date_diff_with_hours)
          ?userid ?courseid
  WHERE{
   ?course elion:courseSource elion:openUniversity.
   ?course elion:courseId ?courseid.
   ?user elion:userId ?userid.
   ?assigment elion:hasCourse ?course.
   ?assigment elion:assignmentAllowSubmissionsFromDate ?fdate.
   ?submission elion:belongAssignment ?assigment.
   ?submission elion:belongsUser ?user.
   ?submission elion:submissionTimeCreated ?timecreated.
  }GROUP BY ?userid ?courseid
}
```

La Consulta 3.5 se utiliza para calcular la diferencia en días entre la fecha de apertura de la tarea hasta la fecha de entrega de los alumnos. También, filtra las tripletas para obtener las pertenecientes a la Open University y agrupa los resultados aplicando la suma de fechas de retraso de un usuario en un curso. Por último, extrae el número total de días.

Todos estos valores de características calculados se escalan en el intervalo $[0, 1]$ para cada asignatura con el fin de homogeneizar los rangos numéricos. Estos valores, así como los del atributo de la etiqueta de calificación, se unen en un conjunto de datos que se utilizará en las tareas de clasificación supervisada. De hecho, el conjunto de datos se divide (aleatoriamente) en subconjuntos de entrenamiento y de prueba con porcentajes de 75 % y 25 %, respectivamente.

Tabla 3.9: Resultados de clasificación de todos los algoritmos utilizados (KNN, DT, SVM, RF, GNB, y MLP) para la predicción de la evaluación continua.

Algoritmo	Exactitud	Clase	Precisión	Sensibilidad.	Valor-F1	Soporte
KNN	0.90	Aprobado	0.86	0.92	0.89	3135
		Suspenso	0.92	0.87	0.89	3441
DT	0.90	Aprobado	0.87	0.92	0.89	3135
		Suspenso	0.92	0.88	0.90	3441
SVM	0.90	Aprobado	0.84	0.95	0.89	3135
		Suspenso	0.95	0.84	0.89	3441
RF	0.90	Aprobado	0.87	0.93	0.90	3135
		Suspenso	0.93	0.87	0.90	3441
GNB	0.89	Aprobado	0.82	0.96	0.88	3135
		Suspenso	0.95	0.81	0.88	3441
MLP	0.90	Aprobado	0.87	0.92	0.90	3135
		Suspenso	0.93	0.87	0.90	3441

Para el modelado de predicciones, se han utilizado una serie de algoritmos de clasificación bien conocidos para comprobar la coherencia de los datos: *"K-Nearest Neighbors"* (KNN), *"Decision Tree"* (DT), *"Support Vector Machine"* (SVM), *"Random Forest"* (RF), *"Gaussian Naive Bayes"* (GNB) y *"MultiLayer Perceptron"* (MLP). Estos métodos se han ajustado mediante validación cruzada de búsqueda exhaustiva (grid search) para el ajuste de los hiperparámetros en la fase de entrenamiento. Con los modelos obtenidos, se realiza un conjunto de predicciones en fase de validación respecto al conjunto de test.

La Tabla 3.9, muestra las métricas obtenidas para todos los clasificadores. Cabe destacar que se alcanzan tasas de acierto cercanas al 90 % para todos los algoritmos, que de hecho muestran métricas de precisión y sensibilidad equilibradas para las dos clases (Aprobado, Suspenso). Por lo tanto, los modelos obtenidos pueden utilizarse para la predicción de las calificaciones de los datos de actividad de nuevos alumnos.

En este sentido, un último paso en este caso de uso consiste en la predicción de las calificaciones en la evaluación continua, pero en este caso con el conjunto de datos de Moodle (Universidad de Málaga), de acuerdo con las características de entrada definidas para este modelo. Como se muestra en la Figura 3.4, las predicciones de las calificaciones resultantes se pueden distinguir visualmente para las dos clases, por lo que podría proporcionar al profesor una herramienta informativa antes de producirse la evaluación final de los estudiantes.

3.4.2. Caso práctico II: Predicción de la calificación final del alumno

De manera similar al caso de estudio anterior, en este caso se entrena un conjunto de modelos de predicción, aunque considerando la calificación final de los alumnos de un curso en función de las interacciones que realizaron en el LMS.

Por tanto, además del conjunto mínimo de características que caracterizan las interacciones de los alumnos (número de envíos, número de visitas y el tiempo de entrega de las tareas), también se incluye en el conjunto de datos la calificación obtenida por cada alumno en la evaluación continua como característica extra para el entrenamiento de los modelos. La calificación final (también Aprobado/Suspenso) se considera como la característica de respuesta que debe predecirse en la fase de prueba.

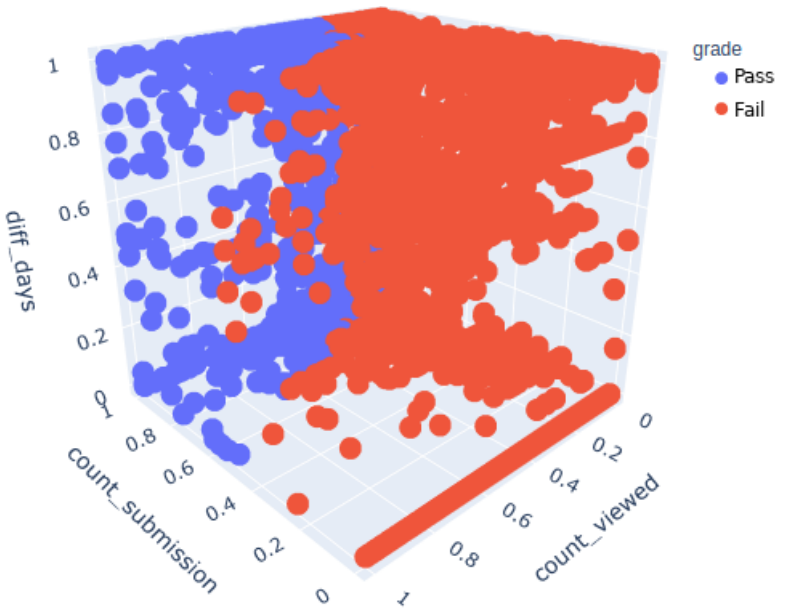

Figura 3.4: Predicción de las calificaciones en evaluación continua con respecto al conjunto de datos de Moodle (Universidad de Málaga).

Fragmento de código 3.6: Obtiene la calificación final por usuario y asignatura.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT ?userid ?courseid ?final_result
WHERE{
  ?course elion:courseSource elion:openUniversity.
  ?enroll elion:inCourse ?course.
  ?enroll elion:enrollmentFinalResult ?final_result.
  ?user elion:isEnrolled ?enroll.
  ?user elion:userId ?userid.
  ?course elion:courseId ?courseid.
}
```

El conjunto de datos resultante se utiliza para alimentar los modelos de clasificación tras dividirlo para entrenamiento (75 %) y validación (25 %). Una vez más, se realiza una validación cruzada de búsqueda exhaustiva para establecer los hiperparámetros de los métodos de clasificación en la fase de entrenamiento. Los valores obtenidos para el conjunto de datos de prueba se muestran en la Tabla 3.10, para todos los algoritmos. En este caso, las precisiones globales son menores que en la evaluación continua, con porcentajes entre el 76 % (obtenido por RF) y el 70 % (obtenido por GNB). Esto se debe principalmente a los bajos valores de precisión y sensibilidad al predecir la clase “Suspenso”, lo que muestra cierto sesgo (probablemente producido por las calificaciones en los exámenes) en la calificación final de los alumnos.

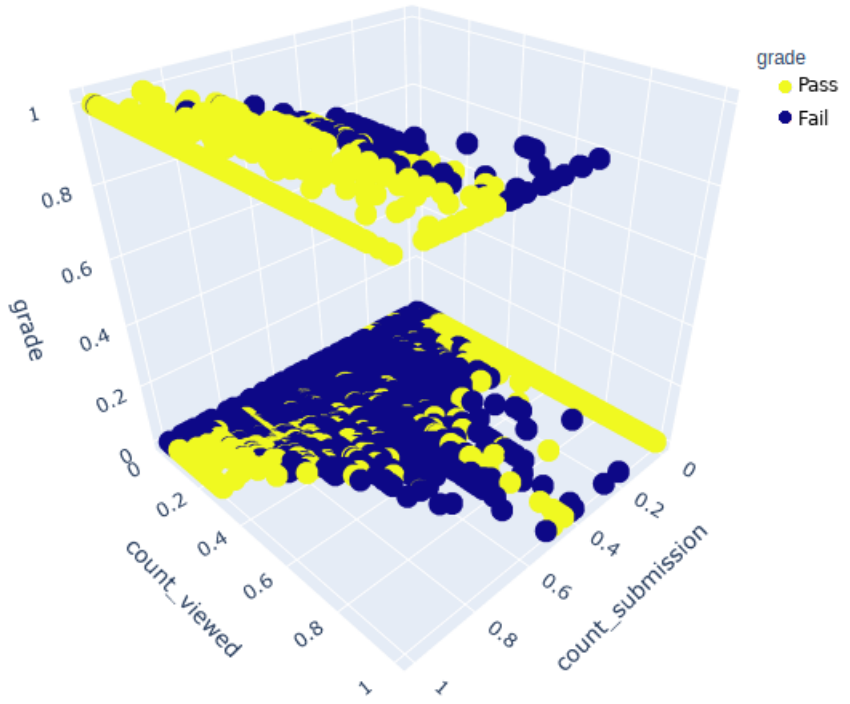

Figura 3.5: Predicción de las calificaciones en la evaluación final realizada sobre el conjunto de datos de Moodle (Universidad de Málaga).

Tabla 3.10: Resultados de clasificación de todos los algoritmos utilizados (KNN, DT, SVM, RF, GNB, y MLP) para la predicción de la calificación final.

Algoritmo	Exactitud	Clase	Precisión	Sensibilidad.	Valor-F1	Soporte
KNN	0.74	Aprobado	0.81	0.92	0.86	5058
		Suspenso	0.51	0.28	0.36	1518
DT	0.74	Aprobado	0.81	0.92	0.86	5058
		Suspenso	0.52	0.27	0.36	1518
SVM	0.74	Aprobado	0.77	0.99	0.87	5058
		Suspenso	0.62	0.03	0.06	1518
RF	0.76	Aprobado	0.85	0.79	0.82	5058
		Suspenso	0.44	0.55	0.49	1518
GNB	0.70	Aprobado	0.80	0.82	0.81	5058
		Suspenso	0.36	0.33	0.34	1518
MLP	0.73	Aprobado	0.79	0.97	0.87	5058
		Suspenso	0.55	0.14	0.22	1518

En la Figura 3.5 se representan las calificaciones finales de los alumnos predichas para el conjunto de datos procedente de la plataforma Moodle de la Universidad de Málaga, donde se visualiza claramente el sesgo producido para la clase "Suspenso". Esto se debe probablemente a un cierto desequilibrio del conjunto de datos resultante debido a un mayor porcentaje de muestras con la

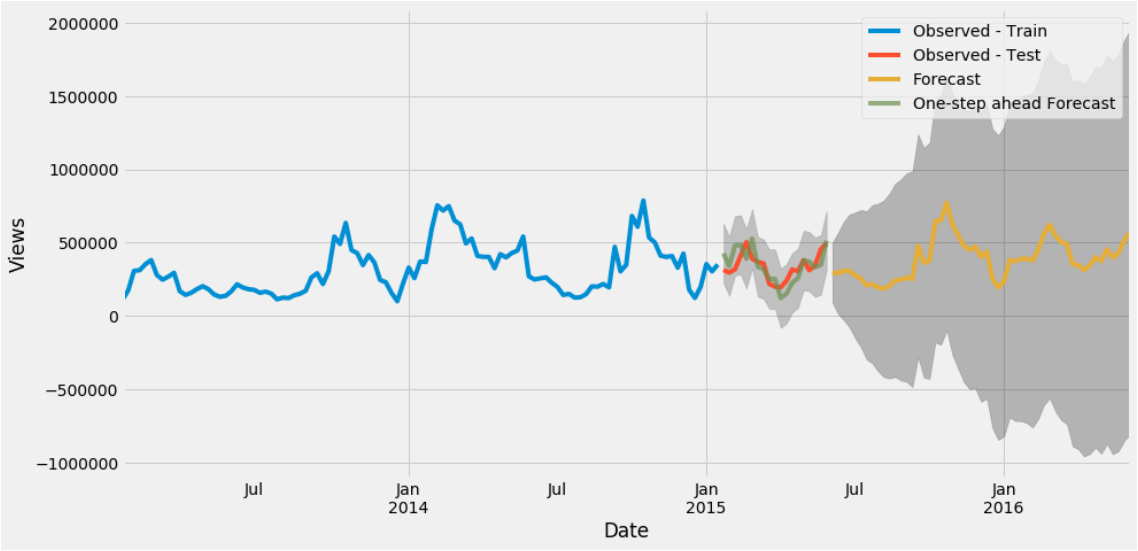

Figura 3.6: Series temporales de las visualizaciones de los alumnos en el LMS, donde se representan los datos observados con respecto a la previsión obtenida por SARIMAX.

etiqueta "Suspenso", que podría mitigarse con un submuestreo hasta alcanzar el equilibrio de clases. Además del moderado rendimiento de la clasificación en este caso de estudio, cabe destacar que la integración de datos de diferentes LMS (Open University) permite generar predictores útiles capaces de reproducir resultados similares en otras fuentes (Universidad de Málaga), por lo que el modelo semántico de sistemas de gestión del aprendizaje en línea aquí propuesto es una contribución útil en esta dirección.

3.4.3. Caso práctico III: Predicción de las visualizaciones de estudiantes mediante series temporales

Otro caso de uso interesante consiste en predecir la tendencia de las visitas de los alumnos en el LMS a lo largo del tiempo, para luego advertir de posibles descensos de actividad en determinados periodos, lo que ayudaría a decidir días concretos del semestre para actualizar contenidos o actividades.

Para ello, un primer paso es obtener los datos necesarios relativos a las visitas que los alumnos realizan en el sistema de gestión del aprendizaje *"online"*, junto con las fechas de dichas visitas. Estos datos se pueden obtener mediante una consulta en SPARQL Consulta 3.7, que agrega el número de clics y filtra las tripletas a las de los datos de la Open University. El conjunto de datos resultante es una serie temporal de visitas agrupadas por semanas para obtener periodos de muestra homogéneos. Para las pruebas de validación se utiliza un subconjunto del 15 % de estos datos.

En este punto, un análisis previo consiste en comprobar si la serie temporal es estacionaria o no, con el fin de decidir qué tipo de algoritmo utilizar (y cómo ajustarlo) para la predicción. Por lo tanto, se utiliza la prueba de Dickey-Fuller sobre el conjunto de datos, lo que da como resultado un intervalo de confianza del 95 %, con un estadístico de prueba de -3,22 y un p-valor de 0,018, por lo que se puede afirmar que la serie temporal es estacionaria.

Para el entrenamiento y la predicción de series temporales se han utilizado dos métodos autorregresivos populares: SARIMAX[15] y Prophet [16]. El primero es una extensión de la clásica modelo autorregresivo integrado de media móvil (ARIMA) que admite la componente estacional, mientras que el segundo aplica un procedimiento de predicción de datos de series temporales basado en un modelo aditivo en el que las tendencias no lineales se ajustan a la estacionalidad anual, semanal y diaria, además de los efectos de las vacaciones. Por lo tanto, estos dos métodos están bien adaptados para caracterizar los distintos periodos de aprendizaje en los cursos universitarios.

Fragmento de código 3.7: Obtener la serie temporal de las visualizaciones

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT (SUM(?numclick) AS ?count_viewed) ?timecreated
WHERE
  { ?x elion:logEduLevel ?edulevel . FILTER (?edulevel = 2)
    ?x elion:logTimeCreated ?timecreated .
    ?x elion:recordCourse ?course .
    ?x elion:logSumClick ?numclick .
    ?course elion:courseSource elion:openUniversity .
}GROUP BY ?timecreated
```

Los hiperparámetros del modelo SARIMAX se han ajustado con un procedimiento de búsqueda por rejilla, eligiendo el que presenta el AIC más bajo (p=P=1, d=D=1, q=Q=0, AIC=412,55). Una vez entrenados los dos modelos, se predice la partición de prueba y se realiza una predicción para las 53 semanas siguientes. Un gráfico ilustrativo de estos resultados se muestra en la Figura 3.6, donde se representan las series temporales obtenidas por SARIMAX con respecto a los datos observados.

Tabla 3.11: Resultados de las métricas de error en las predicciones de series temporales para los modelos SARIMAX y Prophet.

Medida del error	SARIMAX	Prophet
MAE	6.81e+04	6.91e+04
MSE	6.45e+09	6.04e+09
RMSE	8.03e+04	7.77e+04
RMSLE	6.00e−02	5.00e−02

La Tabla 3.11 contiene los resultados obtenidos por SARIMAX y Prophet en términos de métricas de error comúnmente utilizadas para modelos de regresión, donde los datos de prueba observados se comparan con la predicción $(y - \hat{y})$. Estas métricas son: Error absoluto medio (cuyas siglas en inglés son MAE), Error cuadrático medio (MSE), Raíz del error cuadrático medio (RMSE) y Raíz del error cuadrático logaritmico (RMSLE). En general, se puede comprobar que Prophet muestra valores de error más bajos que SARIMAX, aunque ambos métodos se ajustan con éxito con valores de error moderados, por ejemplo, RMSE cercano a 80.000 en 53 semanas (1.132 visualizaciones por semana), con una media de 318.577 visualizaciones de alumnos por semana en los datos observados.

[15]Disponible en la URL `https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html`

[16]Disponible en la URL `https://github.com/facebook/prophet`

3.4.4. Caso práctico IV: Tareas de razonamiento

Para mostrar al máximo el potencial uso del modelo semántico propuesto, este último caso de estudio consiste en la generación de reglas semánticas SWRL para realizar tareas de razonamiento sobre el grafo de conocimiento. Para ello, se ha especificado una clasificación de alumnos en función de sus actividades y del número de trabajos entregados.

Fragmento de código 3.8: Identifica a un estudiante de tipo "Looker".

```
elion:Course(?c)
^ elion:enrollmentNumberOfClicks(?e, ?nclicks)
^ swrlb:greaterThan(?nclicks, ?avgclicks)
^ elion:enrollmentNumberOfSubmissions(?e, ?nsub)
^ elion:courseClicksAVG(?c, ?avgclicks)
^ elion:enrollmentRole(?e, ?r) ^ swrlb:equal(?nsub, 0)
^ swrlb:stringEqualIgnoreCase(?r, "Student")
^ elion:User(?u) ^ elion:isEnrolled(?u, ?e)
^ elion:inCourse(?e, ?c)
-> elion:typeOfUser(?e, "Looker")
```

Las reglas SWRL se han definido utilizando varias propiedades de datos de las clases *Course* y *Enrollement*, esta última conecta a cada usuario con la asignatura en el que está inscrito. Concretamente, para la clase *Course* (Tabla 3.4), se han utilizado las propiedades de datos *CourseClicksAVG* y *CourseNumberOfSubmissions*, que denotan el número medio de clics realizados por los alumnos y el número de entregas declaradas para el curso, respectivamente. Para la clase *Enrollement* (Tabla 3.6), se definen reglas SWRL con respecto a las propiedades de los datos: *enrollmentNumberOfClicks*, para registrar el número de clics que un alumno ha realizado en este curso, y *enrollmentNumerOfSubmissions* que almacena el número de entregas que el alumno ha realizado para este curso. Los valores de estas propiedades se calculan a lo largo de las consultas, por lo que no se almacenan explícitamente a-priori en el repositorio RDF.

La propiedad de datos *TypeOfUser* define, para cada asignatura y alumno, el tipo de alumno según su comportamiento mediante una etiqueta categórica: *Looker* (alto número de clics, pero bajo número de entregas), *Active* (alto número de clics y entregas) y *Passive* (bajo número de clics y entregas). Por tanto, los valores de la propiedad *TypeOfUser* son los inducidos por el razonador (Stardog).

Fragmento de código 3.9: Identifica a un estudiante de tipo "Passive".

```
elion:Course(?c)
^ elion:enrollmentNumberOfClicks(?e, ?nclicks)
^ swrlb:greaterThan(?nclicks, ?avgclicks)
^ swrlb:multiply(?per, ?rate, 100)
^ swrlb:lessThan(?rate, 50)
^ elion:User(?u)
^ elion:courseClicksAVG(?c, ?avgclicks)
^ elion:enrollmentNumberOfSubmissions(?e, ?nsub)
^ swrlb:stringEqualIgnoreCase(?r, "Student")
^ elion:courseNumberOfAssignments(?c, ?nassig)
^ swrlb:divide(?rate, ?nsub, ?nassig)
^ elion:enrollmentRole(?e, ?r)
^ elion:isEnrolled(?u, ?e)
^ elion:inCourse(?e, ?c)
-> elion:typeOfUser(?e, "Passive")
```

De este modo, cada profesor podría analizar los datos de años anteriores para definir reglas de clasificación según sus propios parámetros, es decir, los umbrales numéricos para discriminar entre los comportamientos de los alumnos. Por ejemplo, el fragmento de código de la regla SWRL 3.8 define a un alumno que hace más clics que la media en una asignatura, pero sin realizar entregas, por lo que se le clasifica como *Looker*. La regla SWRL 3.9 denota a un alumno que hace más clics que la media y realiza el 50 % de las entregas requeridas en una asignatura, por lo que se le clasifica como *Passive*. Del mismo modo, el alumno *Active* sería aquel que hace más clics que la media y realiza más del 80 % de las entregas requeridas, en una asignatura determinada.

Fragmento de código 3.10: Tipo de usuario inferido por las reglas SWRL en una asignatura.

```
PREFIX elion: <http://ontologies.khaos.uma.es/e-lion/>
SELECT ?userid ?type WHERE {
  ?user elion:isEnrolled ?e .
  ?user elion:userId ?userid .
  ?e elion:inCourse ?course .
  ?e elion:typeOfUser ?type .
  ?course elion:courseId ?id_course . FILTER(?id_course="66")
  ?course elion:courseSource elion:UniversidadDeMalaga .
}
```

Una vez ejecutadas estas reglas en el razonador, es posible obtener todos los *ids* de los alumnos de un determinado curso junto con sus clasificaciones referentes a su comportamientos. La consulta SPARQL Consulta 3.10 muestra un ejemplo en este sentido, para clasificar alumnos (por *userid*) del curso '66' de la Universidad de Málaga.

3.5. Discusión

A la luz de los casos de uso y resultados anteriores, se puede argumentar que el modelo semántico propuesto es capaz de constituir una base de conocimiento de integración de múltiples fuentes para alimentar eficientemente el análisis y las visualizaciones de datos. Se trata de un claro paso adelante para abordar el reto de la interoperabilidad de datos identificado en las revisiones del estado del arte actuales, por ejemplo, [37, 36], en el contexto de las ontologías para los sistemas de gestión del aprendizaje *"online"*.

Desde una perspectiva práctica, la ontología e-LION puede utilizarse en el núcleo de una estrategia de consolidación de datos abiertos enlazados, en la que los LMS actuales y otras fuentes de datos académicos se consultan sistemáticamente para apoyar a los profesores mediante análisis sobre la evolución del curso. Además, el modelo semántico subyacente resulta útil a la hora de analizar las actividades de los estudiantes en el contexto del rendimiento global de un determinado curso, lo que les proporcionaría una visión global de sus progresos y, por tanto, les conduciría a mejorar sus propios planes de aprendizaje.

En concreto, se pueden extraer una serie de observaciones técnicas en forma de lecciones aprendidas, como se indica a continuación:

- La definición de un modelo semántico, basado en una ontología OWL2, sobre los datos de Moodle, es útil para integrar datos de otras plataformas de gestión de aprendizaje *"online"* o conjuntos de datos educativos. Las ontologías OWL2 permiten la identificación inequívoca de entidades y la identificación de relaciones semánticas que conectan estas entidades. Además, el lenguaje OWL2 proporciona los mecanismos que definen las restricciones lógicas sobre los datos integrados en los casos que sea necesario.

- La ontología OWL 2 propuesta en este trabajo puede alinearse fácilmente con las ontologías y vocabularios del aprendizaje *"online"* existentes gracias a los mecanismos de alineación de ontologías. Por lo tanto, otros datos relacionados con el *"e-Learning"*, es decir, los recursos de los LMS, se pueden consolidar fácilmente dentro de la misma estructura ABox.
- En este sentido, la implementación de un repositorio RDF común que integre datos heterogéneos en un formalismo estándar simplifica las consultas de los usuarios y facilita la obtención de los datos de entrada a los algoritmos.
- El lenguaje SWRL permite definir reglas en el contexto de e-LION, proporcionando así un mecanismo de razonamiento para inferir nuevo conocimiento a partir de los datos integrados, clasificar automáticamente usuarios, etc.

Junto a estos comentarios, cabe mencionar que e-LION podría alinearse con otras muchas ontologías, no sólo en el dominio educativo, sino también en diferentes dominios, como: redes sociales, comportamientos de los usuarios de Covid-19 relacionados con la salud, evolución demográfica y social. Esto permitiría mecanismos más avanzados de integración y consulta de datos y alimentaría el análisis multidimensional.

3.6. Conclusiones

En este capítulo se propone un enfoque semántico para la integración de datos de sistemas de gestión del aprendizaje *"online"* de múltiples fuentes, que comprende la generación de una nueva ontología OWL 2 denominada e-LION. Se definen una serie de funciones de mapeo para consolidar fuentes de datos de diferentes LMS (Moodle, COCO Udemy, Open University) a RDF en un repositorio común, que ahora puede ser utilizado para alimentar análisis avanzados mediante consultas SPARQL, para monitorizar las interacciones de alumnos y profesores. Se ha desarrollado un conjunto de casos de uso para la validación, que abarcan la predicción de calificaciones, la predicción mediante series temporales de las visualizaciones de los estudiantes y el razonamiento semántico con reglas SWRL para la clasificación de los comportamientos de los estudiantes.

Se demuestra que el enfoque semántico propuesto integra adecuadamente los datos de sistemas de gestión de *"e-Learning"*, permitiendo la consulta avanzada y constituyendo un grafo de conocimiento bien fundamentado para mejorar el análisis informativo en el contexto de los LMSs. Esto lleva a la ontología e-LION propuesta a proporcionar un valor científico añadido, que en el contexto del estado del arte actual (como se explica en la Sección 3.2), permite la conexión semántica con otras ontologías y vocabularios relacionados, promoviendo así la generación de amplios datos enlazados en el dominio del aprendizaje *"online"*.

Por lo tanto, la propuesta puede utilizarse en el núcleo de una estrategia de consolidación de datos para futuras aplicaciones, en las que los LMS actuales y otras fuentes de datos académicos se consulten sistemáticamente para apoyar a los profesores con análisis y visualizaciones avanzadas de lo que está ocurriendo en sus asignaturas. Del mismo modo, para los estudiantes, permite analizar las actividades de los alumnos en el contexto del rendimiento global en un determinado curso, lo que les proporcionaría una perspectiva global de su rendimiento en el curso, fomentando así su aprendizaje proactivo.

Como trabajo futuro, se plantea incluir más datos de otros sistemas de gestión del aprendizaje *"online"*, así como actualizar la ontología e-LION para incorporar nuevos atributos relevantes desde diferentes perspectivas del *"e-Learning"*. En este sentido, otra actividad futura es la alineación ontológica de muchas otras no sólo en el dominio del conocimiento educativo, sino también en diferentes dominios, tales como: redes sociales, comportamientos de los usuarios de Covid-19 relacionados con la salud, evolución demográfica y social.

Capítulo 4

Modelo semántico para la integración de datos como soporte a la directiva PSD2 de banca abierta

En los actuales servicios de banca abierta, regulados por la Directiva Europea sobre Servicios de Pago (PSD2)[1] se permite la recogida segura de información de los clientes de servicios bancarios, en su nombre y con su consentimiento, para analizar su situación financiera y sus necesidades. La directiva PSD2 está dando lugar a un número masivo de transacciones diarias entre entidades Fintech, que requieren la gestión automática de los datos implicados, generalmente procedentes de diferentes fuentes y con formatos heterogéneos. En este contexto, uno de los principales retos surge a la hora de definir e implementar esquemas comunes de integración de datos para fusionarlos fácilmente en grafos de conocimiento, permitiendo así la reconciliación de datos y dando lugar a análisis sofisticados. En este sentido, las tecnologías de la Web Semántica constituyen un marco adecuado para la integración semántica de datos que permite enlazar con fuentes externas y mejora la consulta sistemática.

Con esta motivación, en este capítulo se propone un enfoque ontológico para operar como mediador semántico de datos en operaciones de banca abierta del mundo real. De acuerdo con los mecanismos de reconciliación semántica, el grafo de conocimiento subyacente se puebla con los datos implicados en las operaciones de banca abierta PSD2, que se alinean con la información procedente de los sistemas de facturación. En este capítulo, se definen una serie de reglas semánticas para mostrar cómo la clasificación de la solvencia financiera de los clientes y la sugerencia de concepto para transacciónes, pueden inferirse a partir del modelo semántico propuesto.

[1] Disponible en la URL `http://data.europa.eu/eli/dir/2015/2366/2015-12-23`

4.1. Introducción

Los servicios de banca abierta ofrecen hoy en día potentes facilidades en el sector de las Fintech para compartir digitalmente información financiera con terceros, permitiendo a los clientes autorizar sus transacciones a través de interfaces de programación de aplicaciones abiertas y seguras. Este enfoque promueve la rápida transformación digital de los servicios de pago, tradicionalmente gestionados por entidades bancarias, así como la generación de plataformas digitales y empresas que ofrecen a sus clientes servicios de pago y financieros mejorados. En este contexto, la Directiva Europea de Servicios de Pago (PSD2) permite la recopilación segura de información de los clientes bancarios, en su nombre y con su consentimiento, para analizar su situación financiera y sus necesidades [56]. Este nuevo paradigma da lugar a transacciones diarias masivas entre entidades Fintech, que requieren la gestión automática de los datos implicados, generalmente procedentes de diferentes fuentes y en formatos heterogéneos.

En este sentido, iniciativas recientes como el proyecto Helix (AEI-010500-2020-34 NextGeneration EU) y la Estrella European Initiative (IST-2004-027655) [57] se centran en desarrollar las bases fundamentales de una futura plataforma online que democratice el acceso de las pymes a herramientas de financiación y gestión de riesgos. El objetivo de esta plataforma es dinamizar la economía real y la seguridad en las transacciones *"business to business"* (B2B). Hoy en día, estas herramientas sólo están al alcance de las grandes empresas. El objetivo principal en este tipo de proyectos es gestionar de forma integral las operaciones de riesgo comercial con los clientes a través de la posibilidad de contratación online de seguros de crédito por factura, seguros de crédito por deudor, así como el acceso directo a financiación de circulante a través de *Factoring*, todo ello de forma automática e instantánea en una única transacción online, basada en algoritmos predictivos y en el comportamiento histórico en pagos de los diferentes deudores.

Un reto clave en este escenario es definir y aplicar esquemas comunes de integración de datos para fusionarlos fácilmente en grafos de conocimiento. De este modo, se posibilita la conciliación de datos y se da lugar a la generación de análisis sofisticados, por ejemplo, cruzando información entre distintos extractos de cuenta para verificar la coincidencia de los saldos de la empresa, como medida de seguridad adicional contra intentos de fraude.

En este sentido, las tecnologías de la Web Semántica constituyen un marco adecuado para la integración semántica de datos que permite enlazar con fuentes externas y potencia la consulta sistemática. El desarrollo de nuevas ontologías y su uso para la integración de datos está ampliamente documentado en la literatura existente en diferentes dominios de aplicación [28, 29, 30, 31]. En el ámbito financiero y bancario, han ido apareciendo una serie de aproximaciones [57, 58, 59, 60] donde se diseñan ontologías para describir los principales conceptos y relaciones que implican las actividades de gobierno y gestión de datos en este sector. Un estudio representativo en este sentido se ha presentado recientemente en [61].

Concretamente, la conocida *"Financial Industry Business Ontology"* (FIBO) [59] evoluciona desde su publicación inicial en 2014, con un apoyo cada vez mayor a casos de uso relacionados con la gestión de datos sobre valores, informes, análisis y gestión de riesgos. No obstante, aunque todas estas iniciativas constituyen un punto de partida para seguir avanzando, deben tenerse en cuenta aspectos prácticos y normas específicas (como la PSD2) para permitir una integración moderna de datos bancarios abiertos y aplicaciones digitales.

Para hacer frente a este problema, en este capítulo se propone un enfoque ontológico denominado OBO (Open Banking Ontology) para operar como mediador semántico de datos en operaciones reales de banca abierta. Siguiendo este esquema ontológico, se genera un nuevo grafo de conocimiento que se puebla con los datos implicados en las operaciones reales de banca abierta PSD2. Mediante esta propuesta, los datos de las operaciones se alinean con la información procedente de las facturas, según mecanismos de reconciliación semántica. Para ello, se adaptan técnicas de

procesamiento de texto y consultas semánticas para extraer dos corpus de palabras. Después, se comparan con un método de búsqueda difusa para establecer una puntuación de similitud. Esta metodología sigue enfoques similares aplicados con éxito en otros campos como la informática biomédica [62] o el reconocimiento de nombres de empresas que aparecen en noticias de prensa [63].

Además, se trabaja con una serie de reglas semánticas para mostrar cómo la clasificación de solvencia financiera de las entidades y las sugerencias de conceptos de transacción pueden inferirse a partir del modelo semántico propuesto.

Las principales contribuciones de este capítulo se exponen a continuación:

- Se define una ontología utilizando OWL 2 [64] por primera vez para la anotación semántica de las operaciones de gestión de movimientos bancarios y facturas en el contexto de las transacciones PSD2. La propuesta, denominada Open Banking Ontology (OBO), se utiliza como esquema de datos de integración semántica para diferentes fuentes de información que permite la ingesta, consulta y razonamiento de datos de forma estructurada y homogénea.
- De acuerdo con la definición de OBO, también se desarrolla un modelo semántico para la generación de grafos de conocimiento, que se instancian mediante funciones de mapeo específicas para cada fuente de datos diferente. El grafo resultante, compuesto por todas las instancias de datos, se integra en un repositorio RDF, permitiendo la consulta bajo el mismo lenguaje (SPARQL). Se han probado una serie de mecanismos de conciliación con datos de más de 70.000 facturas y 33.000 clientes de banca abierta a través de múltiples operaciones PSD2.
- La propuesta se valida a través de varios casos de uso en el mundo real, incluyendo la conciliación inteligente de movimientos bancarios con sus correspondientes facturas. Además, también se especifica un conjunto de reglas de razonamiento SWRL para clasificar la solvencia financiera de los clientes, incluyendo una generación automática de conceptos que facilitaría la conciliación.

El resto de este capítulo se organiza de la siguiente forma: En la siguiente sección, se incluye una revisión de los trabajos relacionados. En la Sección 4.3, se detalla el diseño de la ontología, incluidos los métodos implementados para construir el grafo de conocimiento. En la Sección 4.4 se explican los mecanismos de reconciliación semántica y los casos de uso de razonamiento. Finalmente, la Sección 4.5 contiene las principales conclusiones y futuras líneas de investigación.

4.2. Trabajos relacionados

En la literatura pasada y actual, no se encuentran muchos estudios que propongan ontologías que traten de los dominios del conocimiento económico y/o financiero. Sin embargo, existen contribuciones importantes que merece la pena destacar.

Un primer intento fue presentado [58], donde se desarrolló una interesante plataforma basada en ontologías para integrar contenidos financieros, bajo una base de conocimiento semántica que proporciona una visión conceptual sobre contenidos de bajo nivel, incluyendo facilidades de búsqueda semántica. Sin embargo, aún constituye una propuesta preliminar para la que no se dispone de la ontología desarrollada y la búsqueda semántica sólo se limita a valores de propiedades concretas.

En 2008, como resultado del proyecto europeo Estrella [2], apareció el *"Legal Knowledge Interchange Format"* (LKIF) [57] como un esquema XML para representar teorías y argumentos (pruebas). Un planteamiento en LKIF consiste en un conjunto de axiomas y reglas de inferencia

[2]ESTRELLA Project `http://www.estrellaproject.org/`

revocables. El lenguaje de individuos, predicados y símbolos de función utilizado puede importarse de una ontología representada en OWL. Al importar una ontología también se importan los axiomas de la misma. También pueden importarse otros archivos LKIF, lo que permite modularizar teorías complejas.

También, en el dominio de los productos financieros, pero con una orientación diferente, se propuso una ontología en 2011 [60] para la anotación de clientes y productos financieros. El objetivo final de este trabajo era modelar un sistema recomendador para vincular productos con perfiles de clientes, por lo que la propuesta se restringió a esta aplicación específica.

Como se comentó en la introducción, una propuesta importante en 2013 fue la Financial Industry Business Ontology (FIBO)[3] [59], que fue concebida como un modelo ontológico compuesto por módulos o sub-ontologías, cada una para un propósito específico en el dominio regulatorio de las finanzas. Esta ontología está alineada con LKIF ya que ambas se centran en la regulación legal de los diferentes actores de las transacciones financieras. Aunque no cubren las directivas sobre movimientos bancarios amparados por la normativa PSD2, podrían alinearse fácilmente con OBO en lo que respecta a clases como *proveedor de servicios*, *persona jurídica* o *organismo regulador*.

Tras esta, en 2017 se propuso el modelo OntoREA [65] para cubrir el modelo UML de contabilidad REA que conceptualiza la lógica económica de la contabilidad en términos de recursos que se intercambian en eventos económicos entre agentes económicos. Se trata de una traducción mediante mapeo utilizando OntoUML [66], aunque no se proporciona ninguna especificación OWL.

Del mismo modo, la ontología COFRIS [67] también se propuso en 2017 para considerar los sistemas de información financiera que modelan los informes contables. También está diseñada con el lenguaje OntoUML, pero con una orientación particular a través de la información financiera de acuerdo con la normativa, por lo que abarca diferentes ámbitos como los movimientos bancarios y métodos de conciliación.

Por lo tanto, aunque estas ontologías describen muchos de los conceptos de primer nivel en el dominio de la regulación económica, representan pasos iniciales en la especificación semántica de mecanismos específicos basados en datos para permitir análisis avanzados de los estados financieros. La ontología OBO que aquí se propone pretende describir un paso más allá en esta dirección, centrándose especialmente en la contabilidad y transacciones bancarias abiertas.

4.3. Modelo semántico propuesto

Esta sección describe la metodología seguida para diseñar la ontología propuesta y ofrece detalles sobre su implementación. A este respecto, también se introducen una serie de componentes y mecanismos para mostrar cómo se construye el grafo de conocimiento subyacente y se puebla con datos bancarios abiertos del mundo real.

A la hora de definir una nueva ontología, una buena práctica es seguir una metodología bien fundamentada que considere todos los aspectos necesarios para la descripción del dominio de conocimiento. En concreto, siguiendo el marco de trabajo común en esta tesis doctoral, se ha utilizado el "Proceso de desarrollo de una ontología 101" [68], para el dominio específico de banca abierta, se organiza en los siete pasos siguientes:

1. *Determinar el dominio y alcance de la ontología.* El dominio de la ontología representa los diferentes actores y la información generada en los servicios de banca abierta. En concreto, se contextualizan los movimientos bancarios y los sistemas de gestión de facturas para permitir las tareas de conciliación de datos.

2. *Considerar la reutilización de ontologías existentes.* Tras revisar la literatura, se han considerado varias ontologías para describir los asientos contables [59]. Sin embargo, ninguna de ellas

[3]EDM Council FIBO `https://spec.edmcouncil.org/fibo`

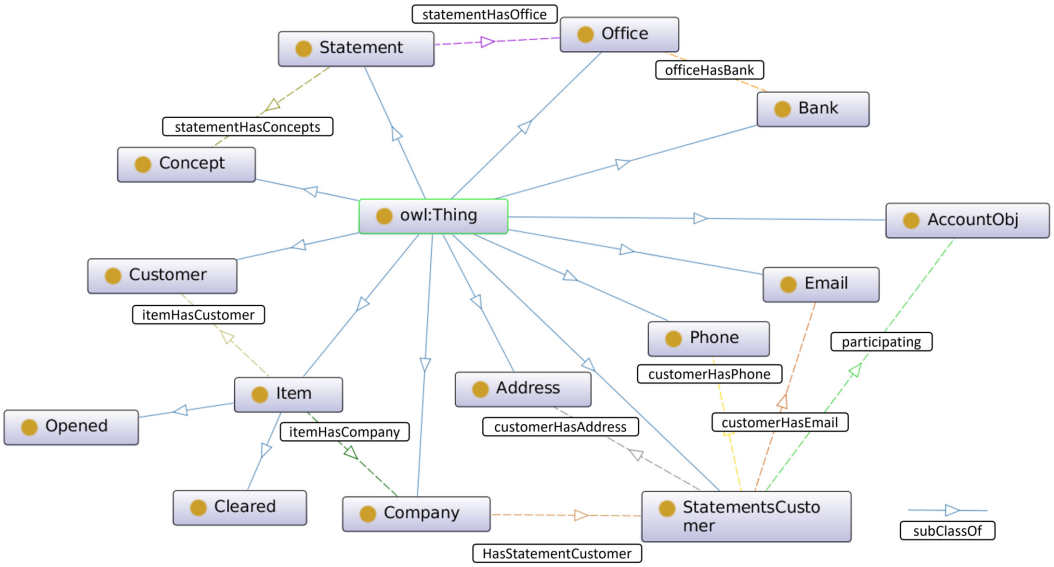

Figura 4.1: Visión general de la Ontología de Banca Abierta (OBO) propuesta.

representa los movimientos bancarios orientados al formato PSD2 y otros aspectos relacionados, como la información contextual de las facturas. Estas ontologías suelen contener una clase *Invoice*, *Receipt* u otros términos similares, pero no describen ontológicamente una factura, con relaciones y componentes. En el enfoque propuesto, el concepto de factura adquiere un papel esencial, ya que permite considerar toda la metainformación que la envuelve para su posterior utilización en mecanismos de conciliación para su vinculación con movimientos bancarios de terceros. Por ejemplo, los productos o servicios a los que se refiere la factura suelen ser desconocidos para la persona jurídica o empresa a la que un determinado cliente ha facturado.

3. *Enumerar términos importantes en la ontología.* Como se ha mencionado en el paso anterior, los términos importantes en OBO corresponden a datos de facturas y movimientos bancarios, que implican transacciones de servicios bancarios abiertos. Ejemplos de estos términos son: *companyId*, *customerId*, *issueDate*, *maturityDate*, *itemCurrencyAmt*, y *documentType*, que están relacionados con la facturación, así como los relacionados con los movimientos bancarios, tales como: *Statement*, *Customer*, *Account number*, *Bank*, *Concept*, y *beneficiary*.

4. *Definir las clases y la jerarquía de clases.* A partir de los términos relevantes, extraemos cuáles de ellos son clases de la ontología. La Figura 4.1 muestra el conjunto de clases que se han definido para describir un servicio de banca abierta. Ejemplos de estas clases son: *Item*, *Statement*, *AccountObj* y *Bank*. En el caso de la clase *Item*, consta de dos subclases, *Opened* y *Cleared*, para especificar si las facturas están pendientes de pago o no, respectivamente.

5. *Define las propiedades de las clases.* Las propiedades de objeto y las propiedades de datos se han creado para conectar instancias de diferentes clases y definir sus atributos. Ejemplos de propiedades de objeto son: *participating*, que conecta un cliente con la cuenta bancaria; *statementHasOffice*, que relaciona un movimiento bancario con la oficina en la que se realizó; y *ItemHasCompany*, que conecta una factura con la empresa. Ejemplos de propiedades de datos son: *statementConcept*, *beneficiary*, *accountNumber itemIssueDate*, *itemAmount*. Las

tablas 4.1, 4.2, 4.3, 4.4 y 4.5 describen detalladamente estas propiedades en la lógica de descripción.

6. *Definir las restricciones de las propiedades.* En este paso se definen las restricciones de cardinalidad y valor. En la ontología propuesta, algunos ejemplos de restricciones son: el rango de la propiedad *itemAmount* tiene que ser un número decimal, el rango de la propiedad *itemIssueDate* tiene que ser una fecha, el *concept* de un movimiento tiene que ser del tipo *string*, etc.

7. *Crear instancias.* Las instancias (o individuos) en el enfoque actual se generan a partir de los documentos JSON que contienen los movimientos bancarios PSD2 reales y los archivos CSV con los datos de las facturas. Estos datos se transforman a RDF mediante funciones de mapeo de acuerdo con las clases y propiedades definidas en OBO, creando a continuación un grafo de conocimiento RDF.

Tabla 4.1: Clase *"Company"*: propiedades de objeto y de datos.

Propiedades de objeto	Lógica descriptiva
hasStastementCustomer	∃ hasStastementCustomer Thing ⊑ Company ⊤ ⊑ ∀ hasStastementCustomer StatementsCustomer
Propiedades de datos	**Lógica descriptiva**
businessName	∃ businessName Datatype Literal ⊑ Company ⊤ ⊑ ∀ businessName Datatype string
companyId	∃ companyId Datatype Literal ⊑ Company ⊤ ⊑ ∀ companyId Datatype int
corporationId	∃ corporationId Datatype Literal ⊑ Company ⊤ ⊑ ∀ corporationId Datatype int
currencyCode	∃ currencyCode Datatype Literal ⊑ Company ⊤ ⊑ ∀ currencyCode Datatype string
fiscalId	∃ fiscalId Datatype Literal ⊑ Company ⊤ ⊑ ∀ fiscalId Datatype string

4.3.1. Modelo Ontológico

La ontología propuesta en este capítulo ha sido desarrollada con Protégé editor[4] tras varias iteraciones de la fase de diseño con expertos en el dominio del conocimiento. Como resultado, la ontología propuesta consta de 14 clases, 10 propiedades de objetos, 35 propiedades de datos y 250 axiomas.

La Figura 4.1 muestra una visión general de la propuesta OBO, por lo que, a continuación se detalla una selección de las principales clases y propiedades. El resto de elementos de OBO pueden examinarse en la documentación, disponible en el enlace `https://ontologies.khaos.uma.es/obo`.

[4]Sitio web Protégé `https://protege.stanford.edu/`

Tabla 4.2: Clase *"Customer"*: propiedades de objeto y de datos.

Propiedades de datos	Lógica descriptiva
businessName	$\exists$ businessName Datatype Literal $\sqsubseteq$ Customer $\top \sqsubseteq \forall$ businessName Datatype string
customerId	$\exists$ customerId Datatype Literal $\sqsubseteq$ Customer $\top \sqsubseteq \forall$ customerId Datatype int
fiscalId	$\exists$ fiscalId Datatype Literal $\sqsubseteq$ Company $\top \sqsubseteq \forall$ fiscalId Datatype string

Tabla 4.3: Clase *"Item"*: Propiedades de objeto y de datos.

Propieades de objeto	Lógica descriptiva
itemHasCompany	$\exists$ itemHasCompany Thing $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemHasCompany Company
itemHasCustomer	$\exists$ itemHasCustomer Thing $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemHasCustomer Customer
Propiedades de datos	**Lógica descriptiva**
indicator	$\exists$ indicator Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ indicator Datatype string
itemAmount	$\exists$ itemAmount Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemAmount Datatype decimal
itemCurrencyAmt	$\exists$ itemCurrencyAmt Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemCurrencyAmt Datatype decimal
itemIssueDate	$\exists$ itemIssueDate Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemIssueDate Datatype dateTime
itemMaturityDate	$\exists$ itemMaturityDate Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemMaturityDate Datatype dateTime
currencyCode	$\exists$ currencyCode Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ currencyCode Datatype string
itemId	$\exists$ itemId Datatype Literal $\sqsubseteq$ Item $\top \sqsubseteq \forall$ itemId Datatype int

1. La clase ***Company*** representa a las empresas que participan en una determinada transacción PSD2. Para esta clase se han definido 5 propiedades de datos: *businessName*, para indicar el nombre de la empresa; *companyId*, que modela el identificador de la empresa; *corporationId* para identificar la sociedad (cuando exista) a la que pertenece la empresa; *currencyCode* que representa la moneda con la que trabaja la empresa y *fiscalId* para almacenar su número de identificación fiscal. Además, la clase *Company* tiene una propiedad de objeto, *hasStatement-Customer*, que vincula la empresa con sus movimientos bancarios PSD2. En la Tabla 4.1 se describen las propiedades de la clase Empresa mediante lógica descriptiva.

2. Se ha incluido la clase ***Customer*** para anotar a los clientes que participan en las transacciones con las empresas. Se ha definido un conjunto de 3 propiedades de datos para esta clase: *businessName*, que representa el nombre del cliente; *customerId*, que almacena el identificador del cliente; y *fiscalId* para establecer el número de identificación fiscal del cliente. La lógica de descripción de estas propiedades se encuentran en la Tabla 4.2.

3. La clase ***Item*** representa las facturas que los clientes deben pagar. Sus principales propiedades se definen en la lógica de descripción en la Tabla 4.3, que comprende 7 propiedades de datos y dos propiedades de objeto. Como propiedades de datos, cabe destacar *itemAmount* que almacena el importe de la factura en la moneda asociada a la empresa correspondiente;

itemCurrencyAmt, para anotar el importe de la factura; *itemIssueDate* que describe la fecha de emisión de la factura; *currencyCode* que indica la moneda en la que está representada la cantidad; e *indicator*, que representa una serie de descriptores asociados a la factura. Las dos propiedades del objeto son: *itemCompany*, que relaciona cada factura con su empresa, y *ItemCustomer*, que relaciona la factura con el cliente.

Tabla 4.4: Clase *"Statement"*: Propiedades de objeto y de datos.

Propiedades de objeto	Lógica descriptiva
statementHasOffice	$\exists$ statementHasOffice Thing $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ statementHasOffice Office
Propiedades de datos	**Lógica descriptiva**
accountNumber	$\exists$ accountNumber Datatype Literal $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ accountNumber Datatype string
statementAmount	$\exists$ statementAmount Datatype Literal $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ statementAmount Datatype int
statementBeneficiary	$\exists$ statementBeneficiary Datatype Literal $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ statementBeneficiary Datatype string
statementCurrency	$\exists$ statementCurrency Datatype Literal $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ statementCurrency Datatype string
statementValueDate	$\exists$ statementValueDate Datatype Literal $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ statementValueDate Datatype dateTime
concept	$\exists$ concept Datatype Literal $\sqsubseteq$ Statement $\top \sqsubseteq \forall$ concept Datatype string

4. La clase ***Statement*** representa los movimientos bancarios realizados sobre una determinada cuenta bancaria. Entre sus principales propiedades de datos, destacan: *statementAmount*, para anotar el importe de la operación; *accountNumber* que indica la cuenta bancaria en la que se ha realizado el movimiento; *statementCurrency* almacena la moneda asociada al extracto; *statementBeneficiary* que representa al beneficiario del movimiento bancario; *concept* y *statementValueDate* que indican el concepto y la fecha valor del movimiento, respectivamente. Para esta clase, se ha definido la propiedad de objeto *statementHasOffice* para asociar cada operación con la sucursal bancaria correspondiente. En la Tabla 4.4 se describen las propiedades de la clase *Statement*.

5. La clase ***Bank*** modela la información de la entidad bancaria. Se han definido tres propiedades de datos para esta clase: *bankGroupId*, que almacena el identificador del grupo bancario al que pertenece el banco; *bankId*, para almacenar el identificador del banco y *bankName*, que almacena el nombre del banco. Esta clase es importante para anotar no solo bancos tradicionales, sino también entidades de banca abierta que participan en transacciones PSD2 con clientes.

6. La clase ***StatementsCustomer*** representa a los titulares de las cuentas bancarias en las que se realizan los movimientos. Para esta clase se han definido una serie de propiedades de datos para almacenar el documento de identidad del cliente, nombre, fecha de nacimiento, dirección, correo electrónico y teléfono, tal y como se describe en la Tabla 4.5. El cliente se relaciona con sus cuentas bancarias a través de la propiedad de objeto *participating*, especificando el rol con el que participa (titular o autorizado).

Tabla 4.5: Clase *"StatementsCustomer"*: Propiedades de objeto y de datos.

Propiedades de objeto	Lógica descriptiva
participating	$\exists$ participating Thing $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ participating AccountObj
Propiedades de datos	**Lógica descriptiva**
customerBirthDate	$\exists$ customerBirthDate Datatype Literal $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ customerBirthDate Datatype dateTime
customerDocument	$\exists$ customerDocument Datatype Literal $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ customerDocument Datatype string
customerNames	$\exists$ customerNames Datatype Literal $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ customerNames Datatype string
address	$\exists$ address Datatype Literal $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ address Datatype string
email	$\exists$ email Datatype Literal $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ email Datatype string
phone	$\exists$ phone Datatype Literal $\sqsubseteq$ StatementsCustomer $\top \sqsubseteq \forall$ phone Datatype string

Tabla 4.6: Software usado en cada etapa.

Etapa	Software	Lenguaje	Descripción	Referencia
Ontología OBO	Protégé	OWL	Diseño de la Ontología	4
Mapeo JSON	RDFLib	Python	Traducción documentos JSON a RDF	5
Mapeo CSV	RDFLib	Python	Traducción de archivos CSV a RDF	5
Repositorio RDF	Stardog	SPARQL	Almacenamiento y consultas	6
Fuzzy Similarity Search	RapidFuzz	Python	Cálculo de similitud entre cadenas de carácteres	8
Algoritmo Genético	jMetalPy	Python	Cálculo de la suma de varios importes	9
Inferencia	Pellet	Java	Razonamiento con la información definida en la ontología	6

4.3.2. Consolidación de datos

Una vez definida la ontología OBO, los datos implicados pueden integrarse y consolidarse siguiendo el modelo semántico definido en un repositorio RDF. La información se almacena en forma de un grafo de conocimiento con un formato homogéneo independientemente de la fuente y evitando inconsistencias semánticas. Con el fin de clarificar la tecnología que se ha utilizado para desplegar el modelo, en la Tabla 4.6 se resume el software y lenguajes que intervienen en cada etapa del proceso. Este proceso se ilustra en la Figura 4.2, según la cual se han implementado

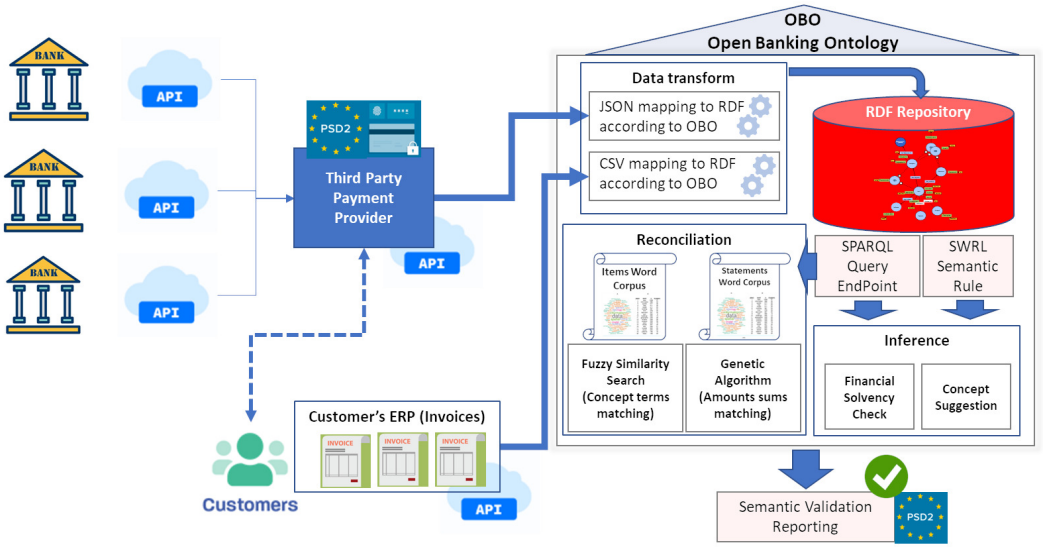

Figura 4.2: Visión general del modelo OBO.

una serie de funciones de mapeo utilizando RDFLib[5] para traducir automáticamente los datos de las facturas emitidas y los movimientos bancarios a formato RDF, poblando así el grafo de conocimiento derivado del esquema OBO.

- Mapeo de datos de facturas en RDF. Estos datos se obtienen de los sistemas ERP de las empresas implicadas en formato CSV, que comprenden cuatro ficheros diferentes: (1) de facturas abiertas, (2) de facturas cerradas, (3) de empresas que emiten facturas a clientes, y (4) de información de clientes, mediante la clase *Customer* en OBO. Además, los contenidos relativos a las clases *Item*, *Company* y *Customer* también se almacenan en el repositorio.

- Mapeo de extractos PSD2 a RDF. Los datos relativos a los extractos bancarios se recogen de las APIs correspondientes (de las entidades bancarias) en formato JSON. El cliente del servicio proporciona al usuario un token de acceso con el que se puede acceder a todos sus movimientos bancarios. Los ejemplos de ficheros JSON en Code Listings 4.1 y 4.2 muestran la información obtenida de los servicios API de las entidades bancarias compatibles con PSD2 relativa a los clientes y sus cuentas bancarias. Las clases *StatementCustomer* y *Statement*, entre otras, están pobladas en forma de instancias semánticas (individuos).

Una vez creadas las tripletas RDF correspondientes a los movimientos bancarios y las facturas, se almacenan en el repositorio RDF, que se ha desplegado mediante una instancia de Stardog[6]. Por lo tanto, los datos integrados pueden consultarse eficazmente a través de un punto de consulta SPARQL. Además, el repositorio RDF está alimentado por un motor de inferencia SWRL para explotar diferentes conjuntos de reglas semánticas diseñadas para realizar tareas de clasificación y validación sobre temas financieros que, junto con los mecanismos de reconciliación semántica, se explican en la siguiente sección.

[5] `https://rdflib.dev/`
[6] `https://www.stardog.com/`

Fragmento de código 4.1: "Customer.json" archivo de ejemplo (anonimizado) al que se accede desde la API de banca abierta compatible con PSD2.


```
{
   "customers":[
      {
         "_id":"A18XXXXXX",
         "address":[
            "LUIS PASTEUR 29071,MALAGA, MALAGA,SPAIN"
         ],
         "document":"A18XXXXXX",
         "birth_date":null,
         "names":"EXAMPLE SA",
         "phones":[
            "9*****500",
            "9*****063",
            "9*****062",
            "9*****010"
         ],
         "accounts_obj":[
            {
               "id":"ESXXXXXXXXXXXXXXXXXXXXX5",
               "participation":"Titular"
            },
            {
               "id":"ESXXXXXXXXXXXXXXXXXXXXX0",
               "participation":"Titular"
            }
         ],
         "emails":[
            "c*****@******.es"
         ]
      }
   ]
}
```


4.4. Validación

En esta sección, se muestra la usabilidad del modelo semántico propuesto mediante la realización de una serie de casos de uso reales orientados a la conciliación de datos y la inferencia semántica. Los datos involucrados, aunque previamente anonimizados, son capturados en el contexto de la mencionada iniciativa de innovación Helix [7].

4.4.1. Caso práctico I: Reconciliación de facturas y movimientos bancarios

Una de las operaciones más habituales en la banca actual, que suele requerir grandes esfuerzos por parte de los expertos, consiste en la inspección (manual) de movimientos para casar pagos parciales relativos a distintos conceptos de las facturas. Una forma de aliviar estos procesos consiste en filtrar aquellas operaciones cuyos conceptos y pagos pueden cotejarse semánticamente de forma automática. Este enfoque puede llevarse a cabo mediante el sistema semántico implementado en este capítulo.

[7]El proyecto de innovación HELIX (`https://www.ongranada.com/proyectos-2/`) está formado por un cluster de entidades financieras que trabajan para maximizar la liquidez de las pequeñas y medianas empresas

Fragmento de código 4.2: "Statement.json" archivo de ejemplo (anonimizado) al que se accede desde la API de banca abierta compatible con PSD2.


```
{
   "statements":[
      {
         "_id":"476XXXX",
         "value_date":"2019-XX-XX",
         "deposit_date":"2019-XX-XX",
         "amount":264XXX,
         "beneficiary":"",
         "bank_reference":"0000XXX",
         "balance":419XXX,
         "currency":"EUR",
         "import_date":"2019-XX-XXT07:30:41+00:00",
         "concepts":[
            "AUTO, S.L.",
            "TRANSFER. EN DIV.",
            "TRANSF.DIVISAS",
            "Origen: CRD",
            "REFERENCIA 204XX"
         ],
         "bank_id":15,
         "bank_group_id":1,
         "account":"ESXXXXXXXXXXXXXXXXXXXX4",
         "bank_name":"Santander",
         "category":null,
         "office":"3XXX"
      }
   ]
}
```


Para la validación, se utiliza datos de 74.000 facturas reales y 150 extractos bancarios, cuyos contenidos se almacenan en el repositorio RDF con las funciones de mapeo explicadas en la Sección 4.3.2. En este sentido, como se muestra en la Figura 4.2, se definen una serie de consultas SPARQL para construir dos corpus de palabras. Uno por cada concepto de factura, junto con los datos de la factura que las empresas suelen almacenar en sus correspondientes ERPs (razón social y identificador fiscal del cliente al que se financió la factura, etc.), y otro por cada *Statement*, se extrae los conceptos de la transacción bancaria (beneficiario, banco, etc.).

La Tabla 4.7 muestra las consultas SPARQL definidas para construir estos corpus con dos ejemplos como resultados. En concreto, el primer conjunto de consultas (primera fila) recupera las propiedades *ItemsConcepts*, *ItemsBusinessName*, *ItemsFiscalId* para obtener la información de los *Items* de dos facturas, mientras que las consultas (segunda fila) se utilizan para seleccionar las propiedades *PSD2Concepts* y *PSD2Beneficiary* de dos *Statements* bancarios de ejemplo.

Una vez construidos los corpus de palabras, se aplica un algoritmo de búsqueda difusa de similitud de cadenas de carácteres [8] [69], para calcular la coincidencia semántica entre las facturas y los extractos bancarios. Este algoritmo se configura con la métrica de distancia de Leverstein y el umbral de confianza de similitud del 80 % para filtrar las coincidencias menos relevantes. En la Tabla 4.8 se muestra un conjunto de cuatro resultados (anonimizados) de alta puntuación obtenidos a partir de un número total de 225 coincidencias (de más de 11 millones de combinaciones posibles).

Además, estas coincidencias filtradas tienen en cuenta la moneda, así como la diferencia entre la fecha de vencimiento de la factura y la fecha del movimiento, que debe ser inferior a un año,

[8] https://maxbachmann.github.io/RapidFuzz/

Tabla 4.7: Consultas SPARQL para la generación de corpus asociados a las facturas y movimientos bancarios.

SPARQL	Resultado

`# Consulta para obtener el corpus de las facturas.`
`PREFIX`
`obo: <http://ontologies.khaos.uma.es/obo/>`
`SELECT ?i ?indicator_field`
`WHERE`
`  {?i obo:indicator ?indicator_field.}`
 	
`SELECT ?i ?indicator_field`
`WHERE`
`  { ?i obo:itemHasCustomer ?indicator.`
`  ?indicator obo:businessName`
`  ?indicator_field`
`  }`
 	
`SELECT ?i ?indicator_field`
`WHERE`
`  { ?i obo:itemHasCustomer ?indicator.`
`  ?indicator obo:fiscalId`
`  ?indicator_field`
`  }`
	`{`
	`  "obo_item00035ed7-8a02-e3dd":[`
	`  "1",`
	`  "1-1",`
	`  "1201****-I",`
	`  "EXAMPLE SL",`
	`  "B2756****"`
	`  ],`
	`  "obo_item0007874d-30a9-a4b0":[`
	`  "1",`
	`  "1902****-F",`
	`  "H-0345****",`
	`  "8771****",`
	`  "EXAMPLE2 SL",`
	`  "****5028G"`
	`  ],`
	`}`

`# Consulta para obtener el corpus de los movimientos.`
`PREFIX`
`obo: <http://ontologies.khaos.uma.es/obo/>`
`SELECT ?s ?concept_field`
`WHERE`
`  { ?s rdf:type PSD2:Statement.`
`  ?s obo:concept ?concept_field.`
`  }`
 	
`SELECT ?s ?concept_field`
`WHERE`
`  { ?s rdf:type PSD2:Statement.`
`  ?s obo:statementBeneficiary`
`  ?concept_field`
`}`
	`{`
	`  "obo_statement06ae705c`
	`  -75f6-4dc0":[`
	`  "EXAMPLE COMPANY S.L.",`
	`  "TRANSFERENCIA INMEDIATA",`
	`  "ref 300XX",`
	`  "BBVAES",`
	`  "Example company"`
	`  ],`
	`  "obo_statement06fdf166`
	`  -8dc6-4cc9":[`
	`  "CON:PAGO FACTURAS",`
	`  "VT. XX-XX-2020",`
	`  "Example company 2, SL"`
	`  ]`
	`}`

ya que no se espera que un cliente tarde más de un año en pagar una factura. Esto se comprueba mediante la consulta SPARQL 4.3, de la que se muestra los de resultados en la Tabla 4.9 compuesta por cuatro nuevos resultados que no se habían obtenido anteriormente.

Por último, en este caso de uso se abordan comprobaciones adicionales para encontrar coincidencias en el caso de pagos parciales, por ejemplo, en aquellas situaciones en las que una factura se paga con varios movimientos bancarios y lo contrario cuando varias facturas se pagan con el mismo movimiento bancario. En estos casos, para calcular las posibles sumas de importes, se ha utilizado un enfoque basado en un Algoritmo Genético (AG) para encontrar las combinaciones óptimas de la suma de subconjuntos. Este algoritmo se ha codificado mediante el framework JMetalPy[9] [70], implementando una variante de AG Elitista con tamaño de población 100, operadores de mutación

[9] https://jmetalpy.readthedocs.io/

Tabla 4.8: Ejemplos de resultados con alta puntuación obtenidos mediante el algoritmo RapidFuzz de similitud difusa de cadenas de caracteres.

URI Item	itemacffa	item55214
URI Statement	statement55d2j	statementd27d4
Item Descriptor	EXAMPLE COMPANY, S.L.	SERVICE FRANCE S.L.U.
Statement Descriptor	A FAVOR DE EXAMPLE COMPANY S.L	SERVICE FRANCE , SL
Score	95	88

Fragmento de código 4.3: Coincidencia importe exacto en factura y movimiento.

```
PREFIX obs: <http://ontologies.khaos.uma.es/obo/>
SELECT ?uri_item ?uri_statement ?amount
       ?date_statement ?date_item
WHERE{
FILTER(?amount = ?amount_item)
  {SELECT ?uri_item ?uri_statement ?amount_item
  (?amount_statement/100 as ?amount) ?date_statement ?date_item
  WHERE{
       ?uri_item obs:itemCurrencyAmt ?amount_item.
       ?uri_statement obs:statementAmount ?amount_statement.
       ?uri_statement obs:statementValueDate ?date_statement.
       ?uri_item obs:itemMaturityDate ?date_item.
       ?uri_item obs:currencyCode ?currency.
       ?uri_statement obs:statementCurrency ?currency.
     }
  }
}
```

SPXCrossover y bit-flip, selección y reemplazo por torneo binario y condición de parada con 25.000 funciones de evaluación de fitness.

En este sentido, un claro ejemplo de resultado se muestra en la Tabla 4.10, donde el AG encuentra una factura cuyo importe (1.039,88 €) coincide exactamente con la suma de los dos movimientos bancarios correspondientes a las cantidades (800 € y 239,88 €). De forma diferente, la Tabla 4.11 contiene un ejemplo de coincidencia en el que el importe de un movimiento bancario (21.913,10 €) se utiliza para pagar dos facturas correspondientes a las cantidades (16.637,50 € y 5.275,60 €). Cabe señalar que, sin la conciliación semántica previa relativa a las descripciones de las facturas y los extractos bancarios, estas correspondencias numéricas serían muy complejas debido al número de combinaciones posibles en el espacio de búsqueda.

4.4.2. Caso práctico II: Inferencia semántica

Este último caso práctico está dedicado a ilustrar dos tareas de inferencia orientadas a permitir la clasificación de clientes y la sugerencia de conceptos mediante razonamiento semántico.

En primer lugar, se calculan los valores de la propiedad de datos *obs:debtRatio* de la clase *Customer* mediante varias consultas SPARQL para cada cliente (*obs:itemHasCustomer*) y empresa (*obs:itemHasCompany*). El ratio de deuda se calcula con la Ecuación 4.1, que se aplica a la agregación de los importes (*obs:itemCurrencyAmt*) de todas las facturas de la clase *Opened* (estos importes son las facturas pendientes), divididos por la suma del importe de todas las facturas (tanto abiertas como cerradas) del cliente.

Tabla 4.9: Resultados obtenidos tras buscar por cantidad exacta.

URI Item	itemb0e44	item40f83
URI Statement	statement14984	statement41d07
Item Descriptor	1, EXP-JCR-100, CLIENT X	1, S07220G, CLIENT Y
Statement Descriptor	PETER, FACT 22000-543	CONTRACT 564, DOSSIER RISK
Amount	241.16	5596.00
Diff Days	-10	20

Tabla 4.10: Resultados de conciliación entre varios movimientos y una factura.

URI Item	item23dcb	item23dcb
URI Statement	statement335ad	statement5eb2b
Item Descriptor	HAPPY FRIENDS, S.A	HAPPY FRIENDS, S.A
Statement Descriptor	HAPPY FRIENDS S.A.	HAPPY FRIENDS
Score	98	95
Item Currency Amt	1039.88	1039.88
Statement Amount	800.00	239.88
Currency Code	EUR	EUR
Diff Days	1	1

$$debtRatio = \frac{\sum amount\ opened}{\sum amount\ cleared + \sum amount\ opened} * 100 \tag{4.1}$$

A continuación, en función de los valores de deuda, se han definido un conjunto de reglas semánticas SWRL para establecer una clasificación de los clientes: La etiqueta *Bajo* se asigna a aquellos clientes con un ratio de endeudamiento inferior al 30 % (Regla 4.4), la etiqueta *Medio* se asigna a aquellos con un ratio de endeudamiento entre el 30 % y el 70 % (Regla 4.5). Por último, la etiqueta *Alto* se asigna a los clientes con un ratio de endeudamiento superior al 70 % (Regla 4.6).

Fragmento de código 4.4: Deuda baja.

```
obs:Customer(?c)
^ obs:debtRatio(?c, ?dRatio)
^ swrlb:lessThan(?dRatio, 30)
-> obs:debtType(?c, "Low")
```

Fragmento de código 4.5: Deuda media.

```
obs:Customer(?c)
^ obs:debtRatio(?c, ?dRatio)
^ swrlb:greaterThanOrEqual(?dRatio, 30)
^ swrlb:lessThan(?dRatio, 70)
-> obs:debtType(?c, "Avarage")
```

Fragmento de código 4.6: Deuda alta.

```
obs:Customer(?c)
^ obs:debtRatio(?c, ?dRatio)
^ swrlb:greaterThanOrEqual(?dRatio, 70)
-> obs:debtType(?c, "High")
```

Tabla 4.11: Resultados de conciliación entre varias facturas y un movimiento.

URI Item	item24b8b	item3069d
URI Statement	statement55d2d	statement55d2d
Item Descriptor	HAPPY FRIENDS S.A	HAPPY FRIENDS, S.A
Statement Descriptor	HAPPY FRIENDS	HAPPY FRIENDS, S.A.
Score	98	95
Item Currency Amt	16637.50	5275.60
Statement Amount	21913.10	21913.10
Currency Code	EUR	EUR
Diff Days	1	1

Por lo tanto, el valor de la propiedad de datos *obs:debtType* es inferido por el razonador en el modelo semántico propuesto, permitiendo así poblar automáticamente esta propiedad en el repositorio RDF. De esta forma, para obtener la clasificación de un determinado cliente en función de su deuda con una empresa concreta, se puede utilizar la consulta SPARQL 4.7 (se refiere al caso concreto del cliente ID 367 con empresa ID 226).

Fragmento de código 4.7: Clasificación crediticia.

```
PREFIX obs: <http://ontologies.khaos.uma.es/obo/>
SELECT ?debtType
WHERE{
?item obs:itemHasCompany ?c. FILTER(?c=company226)
?item obs:itemHasCustomer ?u. FILTER(?u=customer367)
?u obs:debtType ?debtType }
```

En segundo lugar, aprovechando la información almacenada en el grafo de conocimiento, es posible inferir conceptos sugeridos para los diferentes movimientos bancarios. Para ello, se ha definido la regla SWRL 4.8 para fusionar la información de las clases *Opened* y *Customer* que comprenden las propiedades de datos que anotan el número de identificación fiscal del cliente (*fiscalId*), la fecha de vencimiento de la partida (*itemMaturityDate*) y el importe de la partida cerrada (*itemAmount*). A continuación, se utiliza el razonador semántico para concatenar estos valores y formar el valor de la propiedad de datos *suggestedConcept*, con lo que se pueden conciliar los movimientos bancarios.

Fragmento de código 4.8: Concepto sugerido.

```
obs:Opened(?o) ^
obs:Customer(?c) ^
obs:fiscalId(?c, ?fId) ^
obs:itemMaturityDate(?o, ?mDate)
obs:itemHasCustomer(?o, ?c) ^
obs:itemAmount(?o, ?a) ^
swrlb:stringConcat(?con, ?fId, ?mDate, ?a) ->
obs:suggestedConcept(?o, ?con)
```

Un ejemplo de consulta SPARQL para este caso de uso se muestra en la Consulta 4.9, donde dado la URI de un individuo de la clase *Opened*, el razonador es capaz de calcular el valor para la propiedad *suggestedConcept*.

Fragmento de código 4.9: Ejemplo de consulta concepto sugerido.

```
PREFIX obs: <http://ontologies.khaos.uma.es/obo/>
SELECT ?concept
WHERE{
    ?i rdf:type obs:Opened. FILTER(?i = obs:item843)
    ?i obs:suggestedConcept ?concept
}
```

4.5. Conclusiones

En este capítulo, se propone la Open Bank Ontology (OBO) para la anotación semántica y consolidación de los datos involucrados en las transacciones PSD2 entre clientes, bancos y entidades financieras. También se define un modelo semántico para la construcción de grafos de conocimiento y la armonización de datos en el contexto de un proyecto de banca abierta real (Helix), permitiendo la definición de consultas SPARQL y reglas SWRL para el razonamiento. Con fines de validación, también se han desarrollado una serie de mecanismos de mapeo, consultas y algoritmos inteligentes, con especial atención a la conciliación de datos, que pretende permitir la correspondencia automática entre los movimientos bancarios de los pagos de los clientes y las correspondientes facturas anotadas por las empresas en sus sistemas de gestión contable.

Además, también se ha definido un conjunto de reglas de inferencia SWRL de razonamiento para obtener nuevo conocimiento a partir de los datos integrados, con el objetivo específico de clasificar automáticamente a los clientes según sus patrones de deuda y estandarizar los conceptos sugeridos para las facturas.

Cabe destacar que la ontología propuesta constituye el primer intento de modelar los movimientos y actividades bancarias con especial atención al estándar europeo PDS2. Por lo tanto, se esperan nuevas extensiones de este trabajo, incluyendo la federación con otros grafos de conocimiento relacionados en el dominio de las aplicaciones y estándares de las Fintech.

En este sentido, como trabajo futuro, está previsto integrar más datos de diferentes sistemas de gestión de facturas y actualizar la ontología OBO para incorporar nuevos atributos relevantes desde diferentes perspectivas, tales como: opiniones en redes sociales, rasgos de comportamiento de la empresa en sus relaciones comerciales con los clientes, etc. Estos nuevos conocimientos permitirán realizar nuevos análisis, teniendo en cuenta más factores y actores.

Capítulo 5

Marco de trabajo para la gestión de la seguridad en bases de datos orientadas a grafos

Las bases de datos basadas en grafos están diseñadas con los objetivo de mejorar el rendimiento y la flexibilidad. La mayoría de los enfoques existentes para diseñar bases de datos NoSQL seguras se limitan a la fase final de implementación, evitando la fase de diseño inicial relativa a cuestiones de seguridad y control de acceso en niveles de abstracción superiores. Garantizar la seguridad y el control de acceso en las bases de datos basadas en grafos resulta complejo, ya que cada enfoque difiere significativamente en función de la tecnología de que se trate.

Con esta motivación, en este capítulo se propone el primer marco independiente de la tecnología para diseñar bases de datos seguras basadas en grafos. Nuestra propuesta eleva el nivel de abstracción modelando la base de datos y los requisitos de seguridad al mismo tiempo mediante ontologías, apoyadas por la plataforma TITAN [4], lo que nos permite abordar ambos aspectos de forma sencilla. Por lo tanto, las grandes ventajas del enfoque propuesto son: permitir a los diseñadores de bases de datos centrarse en la seguridad y en los datos que deben protegerse, ignorando los detalles de implementación; facilitar el diseño seguro y la migración rápida de las reglas de seguridad derivando en medidas de seguridad específicas para cada tecnología subyacente; y permitir a los diseñadores de bases de datos comprobar mediante el razonamiento ontológico si las reglas de seguridad son coherentes. Para demostrar la aplicabilidad de la propuesta, la aplicamos a un caso práctico basado en el control de acceso a los datos de un hospital.

5.1. Introducción

Las bases de datos NoSQL se han convertido en la piedra angular de múltiples procesos empresariales gracias a su enfoque en el rendimiento y el alto volumen de datos. Desde las redes sociales hasta la detección de fraudes, las bases de datos NoSQL permiten la escalabilidad de las aplicaciones sobre grandes volúmenes de datos que, de otro modo, sería imposible explotar utilizando tecnologías relacionales. Por desgracia, este rendimiento se ha conseguido a costa de otras características. Como consecuencia, la seguridad y la privacidad han quedado relegadas a un segundo plano [71, 72] a pesar de las potenciales pérdidas económicas y reputacionales derivadas de las fugas de información.

Entre los diferentes tipos de bases de datos NoSQL podemos encontrar: i) Las basadas en clave-valor, como DynamoDB o Redis, donde se accede a los datos utilizando claves únicas, ii) las basadas en datos columnares, como Cassandra o HBase, donde las claves están compuestas por combinaciones de columna, filas y marcas de tiempo, iii) las bases de datos orientadas a documentos, como MongoDB o CouchDB, donde la información se almacena como documentos en YAML, y iv) las bases de datos orientadas a grafos, como Neo4J o GraphBase, que son ampliamente conocidas por tener un rendimiento superior a las anteriores cuando acceden a millones de datos (nodos). Para complicar aún más el diseño seguro de las bases de datos NoSQL, incluso las tecnologías que comparten el mismo tipo y modelo de base de datos NoSQL, presentan diferencias en su implementación y en los mecanismos de seguridad que ofrecen.

En la última década, tanto la industria como el mundo académico han acordado que la seguridad debe incorporarse desde las primeras fases de desarrollo siguiendo lo que se hace llamar seguro por diseño [73], combinando principios de la seguridad y de la ingeniería de software. Esta filosofía entra en conflicto con el estado actual del diseño de bases de datos NoSQL. Actualmente, hasta donde sabemos, no existe ninguna metodología o marco sistemático bien conocido que apoye al diseñador de bases de datos NoSQL en el diseño seguro a niveles de abstracción más altos. Por lo tanto, el diseñador de la base de datos no sólo tiene que mantener un profundo conocimiento de la tecnología de la base de datos subyacente, sino también aprender los mecanismos de seguridad concretos para aplicar correctamente todas las cuestiones de seguridad y controles de acceso necesarios desde las primeras etapas.

Para abordar este problema, en este capítulo se propone el primer marco de diseño de bases de datos NoSQL. Nuestra propuesta permite la extensión de conceptos de bases de datos NoSQL con ontologías más específicas, que capturan la idiosincrasia de cada tipo de base de datos. A su vez, estas extensiones nos permiten proporcionar capacidades de derivación automática, facilitando la implementación de diseños de bases de datos seguras de forma rápida y sin errores.

Más concretamente, en este capítulo nos centramos en el diseño de una ontología para bases de datos basadas en grafos y mostramos el proceso completo desde el diseño hasta la implementación. Cabe señalar que para abarcar todos los almacenes de datos, es necesario desarrollar ontologías específicas, cada una de las cuales cubra las particularidades de cada tipo de almacén de datos, por lo que queda fuera del alcance de este capítulo.

Con el fin de mostrar la amplia escalabilidad de nuestro enfoque, a lo largo de este capítulo se hace hincapié en la independencia de la tecnología final de implementación de nuestra propuesta. Además, también mostramos la aplicabilidad de nuestro enfoque desarrollando un caso de estudio completo sobre una base de datos orientada a grafos, incluyendo comprobaciones de modelos y reglas de transformación. Hasta donde sabemos, este es el primer trabajo que muestra cómo los mecanismos de seguridad y control de acceso considerados en la fase de diseño, pueden derivarse de forma semiautomática a la implementación final, evitando así detalles de implementación en la fase de diseño.

Además, la integración de nuestro enfoque con TITAN nos permite proporcionar razonamiento

y analítica avanzados a la organización que implementa la base de datos. TITAN es una plataforma software para gestionar todo el ciclo de vida de los flujos de trabajo de ciencia de datos, desde el despliegue hasta la ejecución en el contexto de aplicaciones Big Data. TITAN utiliza la semántica para tratar los datos y crear componentes interoperables, mejora los procesos de análisis de datos y garantiza la reutilización y el acceso eficientes a los componentes software. Como tal, la base de datos NoSQL diseñada con nuestro enfoque no sólo implementa la seguridad por diseño, sino que también ofrece un acceso más fácil a otros componentes relevantes como algoritmos y aplicaciones gracias a TITAN. El núcleo de TITAN es la ontología BIGOWL [52]. BIGOWL define el conjunto de metadatos para anotar flujos de trabajo analíticos de Big Data, incluyendo componentes que ejecutan algoritmos, fuentes de datos, restricciones de operación y planificación de ejecución.

Con el fin de integrar nuestro marco de diseño de bases de datos NoSQL en TITAN, BIGOWL se amplía para permitir la descripción semántica, tanto de la estructura de la base de datos NoSQL, como de las políticas de seguridad. Se crean dos nuevas clases principales, *Database* y *SecurityRule*. *Database* es una subclase de la clase *Data*, y está relacionada con la clase *SecurityRule*. A continuación, se definen las clases y propiedades de la ontología que describen una base de datos orientada a grafos junto con sus elementos de seguridad específicos.

Las grandes ventajas de nuestro marco de trabajo son:

- El diseñador puede incorporar la seguridad desde el principio, utilizando una capa de alta abstracción que oculta los entresijos de determinadas tecnologías.
- El diseñador puede retrasar la selección de la tecnología de base de datos hasta que los requisitos de diseño estén claros.
- Nuestro enfoque incorpora la capacidad de derivar la implementación para tecnologías objetivo específicas, ahorrando así tiempo, evitando errores de implementación y liberando al diseñador de saber cómo debe implementarse una política de seguridad específica en una tecnología objetivo, si es que de alguna manera es posible.
- La incorporación de una capa de ontología que nos permite soportar capacidades de razonamiento, ofreciendo al diseñador un análisis de posibles inconsistencias o del grado de accesibilidad de cierta información.

Nos gustaría señalar aquí que, hasta donde sabemos, nuestro enfoque es el primero en aprovechar las grandes ventajas del razonamiento sobre ontologías para evitar ambigüedades y facilitar la generación automática de los detalles de implementación. Además, las ontologías se han construido siguiendo la metodología FAIR (Findability, Accessibility, Interoperability and Reusability) [74].

Los principios FAIR tienen por objeto mejorar la localización, la accesibilidad, la interoperabilidad y la reutilización de los datos producidos. Estos principios conllevan definir un estandar de comunidad sobre los metadatos necesarios para que los datos sean localizables y accesibles de forma automática por las máquinas. Una vez que las máquinas acceden a los datos, los pueden analizar y reutilizar. Dado que las ontologías suelen ser el resultado de actividades de investigación y contienen definiciones lógicas del dominio de la investigación. Las ontologías deben tratarse como una parte esencial de cualquier conjunto de datos y aplicárseles los principios FAIR. Para evaluar la conformidad de nuestra ontología con los principios FAIR hemos utilizado el validador FOOPS! [75], un servicio web para detectar las mejores prácticas según cada principio FAIR, obteniendo la máxima puntuación de 17 sobre las 24 pruebas.

Para demostrar la validez de nuestra propuesta, la hemos aplicado a un caso de uso centrado en el ámbito sanitario. Este ámbito es ideal debido a los aspectos de seguridad que deben tenerse en cuenta, especialmente en lo que respecta a la accesibilidad de la información sensible que puede obtenerse tanto directa como indirectamente. Lamentablemente, y con el fin de evitar un capítulo

muy extenso, no se ha podido incluir todo el material utilizado. Las versiones completas de todo el material utilizado en este capítulo, como la ontología, el metamodelo de la base de datos orientada a grafos, las transformaciones del modelo, etc., pueden consultarse en el github de la organización Aether[1].

El resto de este capítulo se estructura de la siguiente manera: La Sección 5.2 presenta los antecedentes y trabajos relacionados sobre ontologías y seguridad en bases de datos NoSQL. La Sección 5.3 detalla los aspectos clave necesarios para implementar políticas de seguridad en bases de datos orientadas a grafos. La Sección 5.4 presenta el marco de trabajo propuesto para el diseño seguro de bases de datos NoSQL y sus componentes. La Sección 5.5 presenta la integración de nuestro marco en TITAN [4], la plataforma ontológica para la interoperabilidad entre componentes de Big Data. La Sección 5.6 presenta el caso de estudio centrado en el ámbito sanitario. Finalmente, la Sección 5.7 presenta las conclusiones y esboza los trabajos futuros.

5.2. Trabajos relacionados

En esta sección, se presentan los trabajos relacionados con las propuestas actuales que tratan de modelar las cuestiones de seguridad en las primeras fases del diseño. Además, se realiza una revisión bibliográfica sobre la combinación de ontologías, bases de datos y aspectos de la seguridad.

Aunque en la mayoría de los desarrollos la seguridad se considera un aspecto importante, es muy habitual que se aborde al final del proceso de desarrollo, añadiendo las restricciones de seguridad necesarias sobre un sistema que, al estar ya implantado, presenta poca flexibilidad al cambio. El concepto de seguro por diseño trata de incorporar las necesidades de seguridad lo antes posible en el proceso de desarrollo, de forma que se tengan en cuenta en el diseño y en las posteriores decisiones de implementación, desarrollando sistemas finales más fiables [76].

En este sentido, podemos encontrar propuestas enfocadas al desarrollo de sistemas seguros que utilizan notaciones para la especificación de requisitos y diseño (como UML, i*, etc.) y las extienden para representar aspectos de seguridad, o bien metodologías y procesos de desarrollo completos que han sido enriquecidos con actividades de seguridad [77, 78, 79, 80]. Son especialmente interesantes las que aplican el enfoque de desarrollo dirigido por modelos, ya que permiten la obtención automatizada de modelos para plataformas específicas e implementaciones finales en herramientas específicas, considerando los aspectos de seguridad definidos inicialmente y ahorrando tiempo y costes de desarrollo [81, 82].

El desarrollo de bases de datos NoSQL, como cualquier otro sistema software, contempla las etapas de especificación de requisitos y diseño, pero teniendo en cuenta las particularidades de este tipo de sistemas. En la literatura encontramos propuestas que abordan el diseño de un tipo de base de datos NoSQL (columnar, documental, etc.) [83, 84, 85, 86] o que intentan alcanzar un mayor nivel de abstracción con conceptos comunes a cualquier tecnología NoSQL [87, 88, 89]. Sin embargo, las propuestas para el diseño de bases de datos NoSQL dejan de lado la incorporación de requisitos no funcionales como la seguridad, relegándola a las etapas finales de implementación, lo que se traduce en sistemas menos seguros y fiables [73, 90].

A nivel de implementación encontramos limitaciones para definir aspectos de seguridad en herramientas de gestión de bases de datos NoSQL [91, 92, 93] y trabajos que aportan avances para tecnologías NoSQL específicas como las documentales [94] o columnares [95] o para herramientas de gestión de bases de datos específicas, como MongoDB o Cassandra [96].

Para bases de datos orientadas a grafos encontramos trabajos centrados en sistemas basados en control de acceso, en [97] se propone un sistema basado en reputación, [98] que trabaja en seguridad de grano fino o [99], donde se extiende Neo4J con un plugin para definir políticas basadas en usuarios pero sin considerar roles ni reglas de grano fino.

[1] Organización Aether `https://github.com/ProyectoAether`

Nuestro trabajo previo ha tratado la seguridad por diseño en bases de datos NoSQL pero centrándose en las orientadas a documentos. Se ha propuesto como punto central un metamodelo a nivel de diseño que incluye aspectos estructurales y de seguridad específicos de las bases de datos orientadas a documentos, a partir del cual se puede generar una implementación de MongoDB de forma automatizada [100]. Esto se complementa con un enfoque de modernización que utiliza ontologías de dominio para analizar los datos con el fin de detectar información confidencial y hacer recomendaciones para que el diseñador/a añada las políticas de seguridad necesarias para protegerla [101].

La combinación de ontologías y bases de datos ha sido ampliamente estudiada desde diferentes perspectivas en las últimas dos décadas. Los trabajos [102] y [103] presentan estudios sobre el alineamiento de contenidos de bases de datos relacionales a ontologías para representar formalmente el contenido de la base de datos y explotar las capacidades de razonamiento de la ontología. En 2022, [104] introduce el aprendizaje de ontologías a partir de bases de datos relacionales como una oportunidad para la integración semántica. Siguiendo la misma filosofía de los trabajos anteriores, [105] y [106] alinean el contenido de una base de datos MongoDB a una ontología, mientras que [107] propone un sistema de integración semántica basado en ontologías para un almacén de datos NoSQL orientado a columnas. Por último, trabajos más recientes mapean bases de datos orientadas a grafos a ontologías [108] y [109]. Todos estos enfoques pretenden representar el contenido de la base de datos como una ontología. Hasta donde sabemos, ninguna ontología representa el esquema o la estructura de las bases de datos. La propuesta descrita en este capítulo se basa en dicha representación.

Por otra parte, el uso de ontologías en ciberseguridad es un área destacada. Concretamente, el uso de ontologías se ha propuesto como solución para diversas tareas, desde modelar ciberataques hasta facilitar el trabajo de auditores o analistas. En este sentido, OBAC [110] utiliza conceptos y relaciones de la ontología de dominio de un conjunto de datos para el acceso seguro a datos FAIR y [111] propone el uso de tecnologías de web semántica para implementar el control de acceso en entornos multidominio. Otro ámbito de investigación atractivo es el de la evaluación de la seguridad. La revisión bibliográfica más reciente sobre este tema es [112]. Según este estudio, la mayoría de las ontologías identificadas tienen como objetivo describir los campos de la seguridad del software y las pruebas de software, incluidos sus diversos subdominios, por ejemplo, gestión de riesgos, políticas de seguridad, análisis de incidentes, patrones de ataque, pruebas de rendimiento, pruebas de sistemas expertos, etc.

Por último, los modelos de control de acceso para la autorización de sistemas, como RBAC (Role-Based Access Control) y ABAC (Attribute-Based Access Control), no han sido ajenos al auge de las tecnologías de web semántica. En [113] se propone una ontología para extender XACML [114], que es un XML para soportar RBAC. ROWLBAC [115] pretende aportar formalismo a los lenguajes de políticas modelando RBAC en OWL. En ROWLBAC, las entidades (Usuarios, Roles, Acciones, etc.) se representan como clases OWL. Un intento de modelar conceptos ABAC en OWL se lleva a cabo en [116]. En este caso, se incluyen elementos complejos de OWL, como clases disjuntas, para enriquecer la ontología.

Hasta donde sabemos, la propuesta presentada en este capítulo, complementa todas las anteriores, ya que se basa en una ontología que permite a los diseñadores representar formalmente la estructura de las bases de datos NoSQL, junto con las políticas de seguridad necesarias.

El uso de una ontología permite representar explícita y formalmente, tanto el esquema de la base de datos como las políticas de seguridad, lo que reduce la ambigüedad y proporciona un marco unificador para los desarrolladores de bases de datos. Además, esta representación permite que el esquema sea procesado por un ordenador, proporcionando un mecanismo para automatizar el proceso de creación de la base de datos. Aunque el problema de modificar el esquema de la base de datos o definir una nueva regla de seguridad supone una limitación, no es necesario ajustar la ontología, sino sólo añadirle más individuos. Implementar nuestra propuesta como un flujo de

trabajo de la plataforma TITAN minimiza el impacto de llevar a cabo una modificación de este tipo, ya que el proceso estaría totalmente automatizado.

Tal y como se ha argumentado anteriormente, y con el fin de mostrar la aplicabilidad y escalabilidad de la ontología propuesta, en este trabajo se diseña y se isntancia la ontología para bases de datos orientadas a grafos junto con las políticas de seguridad que se le pueden aplicar. Hasta donde sabemos, no hemos sido capaces de encontrar en la literatura una ontología con características similares.

5.3. Políticas de seguridad en bases de datos orientadas a grafos

Al abordar la seguridad en bases de datos NoSQL desde el punto de vista del diseño, necesitamos por un lado los aspectos estructurales propios del tipo de base de datos (orientada a grafos, a documentos o a columnas) y por otro la definición de políticas de seguridad relacionadas con ellos, que puedan ser independientes y reutilizables para diferentes tipos de bases de datos NoSQL.

Para ayudar en el diseño de políticas de seguridad sobre bases de datos orientadas a grafos, presentamos dos metamodelos: uno con los conceptos estructurales de las bases de datos orientadas a grafos (Figura 5.1) y otro con los elementos necesarios para definir políticas de seguridad sobre bases de datos, independientemente de su tipo (Figura 5.2). El metamodelo para bases de datos orientadas a grafos permite establecer todos los elementos estructurales necesarios, de forma que una base de datos orientadas a grafos tiene elementos que pueden ser nodo o relación entre dos nodos. Tanto los nodos como las relaciones pueden tener asociado un campo con un tipo de datos.

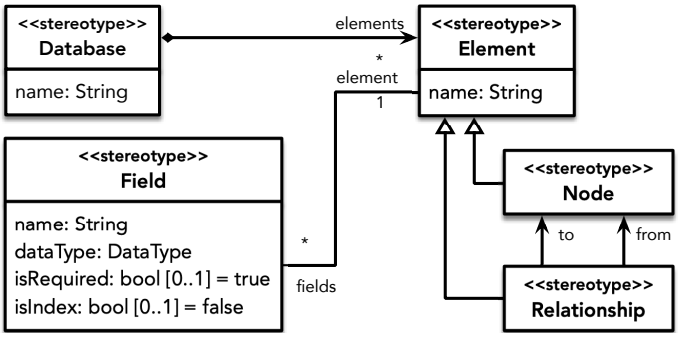

Figura 5.1: Metamodelo de la parte estructural de una base de datos orientada a grafos.

Por otro lado, el metamodelo de seguridad permite establecer políticas de seguridad sobre la base de datos. Para ello, se permite definir un conjunto de reglas de seguridad que conceden o deniegan privilegios (Crear, Leer, Actualizar, Eliminar) sobre elementos de la base de datos a determinados usuarios. Para cada privilegio implicado en la regla, se puede especificar una condición que debe cumplirse.

Para clasificar a los usuarios de la base de datos, se utiliza una política de control de acceso basada en roles para definir jerarquías de roles y asociar usuarios a dichos roles. Esta política de control de acceso es la más utilizada y soportada por las herramientas finales. Las reglas de seguridad pueden asociarse a elementos de la base de datos (SecurityRuleElement) o a campos de dichos elementos (SecurityRuleField). En este capítulo, nos centramos en bases de datos orientadas a grafos, estas reglas se refieren a nodos y relaciones o a atributos de los mismos. Sin embargo, la notación de seguridad es independiente del tipo de base de datos NoSQL, siendo aplicable a otros

tipos (como las documentales y las columnares) refiriéndose en ese caso a los elementos específicos de ese tipo de base de datos (documentos, columnas, etc.).

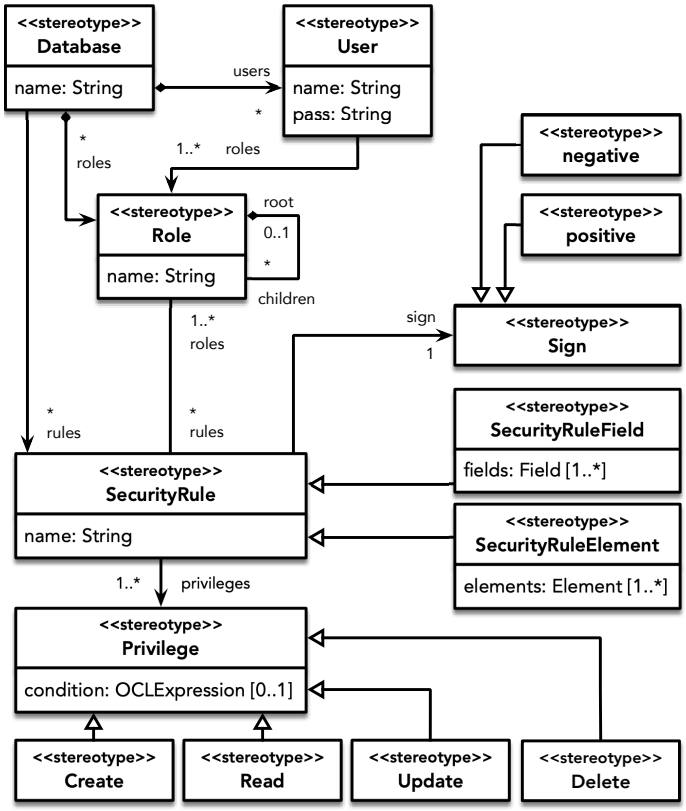

Figura 5.2: Metamodelo de la parte de seguridad de una base de datos orientada a grafos.

5.4. Modelo propuesto

Para cubrir el objetivo principal de este capítulo, primero necesitamos definir la ontología necesaria que englobe y represente genéricamente el dominio de la seguridad en bases de datos orientadas a grafos. La ontología propuesta enriquece y facilita la integración de la información con las plataformas existentes y futuras. En este sentido, BIGOWL se extiende para modelar tanto la estructura (Figura 5.1) como las reglas de seguridad (Figura 5.2) de una base de datos orientada a grafos. En concreto, se ha creado una subclase de la clase BIGOWL *Data* llamada *Database*, que contiene una subclase llamada *NoSQLDatabase*. Por último, se incluye la clase *GraphsDatabase* como subclase de *NoSQLDatabase*. Cabe destacar aquí la escalabilidad del enfoque propuesto a cualquier base de datos NoSQL añadiendo la subclase correspondiente que recoja los requisitos específicos de cualquier base de datos.

Para el diseño de la ontología de seguridad en bases de datos orientadas a grafos, de acuerdo a la metodología adoptada en esta tesis, se ha seguido el proceso de desarrollo "Ontology 101" [117], para el dominio específico tratado en este capítulo. El proceso consta de los siguientes siete pasos:

1. *Determinar el dominio y alcance de la ontología.* El dominio de aplicación de la ontología es la definición de los aspectos relacionados con la seguridad y la estructura de los elementos que definen las bases de datos orientadas a grafos.

2. *Considerar la reutilización de ontologías existentes.* Como se ha mencionado, se considera la reutilización e integración con la ontología BIGOWL[52]. Esta ontología define las clases *PrimitiveType* y *StructuredType* que tienen como padre a la clase genérica *DataType*; El tipo de un campo es entonces un *DataType* de BIGOWL.

3. *Enumerar términos importantes de la ontología.* Los términos esenciales de nuestra ontología han sido analizados y extraídos de los metamodelos presentados en la Sección 5.3. Ejemplos de estos términos son: *SecurityRule*, *Privilege*, *Role*, *User*, *ruleSign*, *RoleHasRules*, *UserHasRole*, etc. Por lo tanto, también definiremos términos genéricos de bases de datos orientadas a grafos, como: *Node*, *Relationship*, *Field* y *DataType*.

4. *Definir las clases y la jerarquía de clases.* Las clases de la ontología se extraen de los términos relevantes. Se definen las clases y su jerarquía al nivel de detalle necesario para clasificar a los individuos. La Figura 5.3 muestra las clases de la ontología. Las clases de BIGOWL están marcadas con color naranja. Las clases marcadas en verde modelan la estructura de una base de datos orientada a grafos, mientras que las azules representan los elementos de seguridad. Las figuras muestran que algunas de estas clases están relacionadas mediante propiedades de objeto o la relación subclase.

5. *Definir las propiedades de las clases.* Se han definido las propiedades de objetos y datos necesarias para representar las conexiones requeridas entre los diferentes individuos y almacenar la información respectivamente. Ejemplos de propiedades son: *DatabaseHasRole* que indica los roles que existen en la base de datos, *UserHasRole* relaciona los usuarios con los roles que tienen, *ruleSign* almacena el signo de la regla, *SecurityRuleDefineElements* que relaciona la regla de seguridad con los elementos sobre los que se aplica. En la Tabla 5.1 se describen las propiedades ontológicas de las clases relativas a la parte de seguridad en formato de lógica de descripciones (definido en la Tabla 2.1). Ejemplos de propiedades para la parte estructural de las bases de datos orientadas a grafos son (ver Tabla 5.2): *hasElements* que relacionan la base de datos con los elementos que contiene, *hasRelationTo* y *hasRelationFrom* para conectar una relación con los nodos, *hasElementField* para asignar un campo a un elemento, etc.

6. *Definir las restricciones de las propiedades.* Este paso incluye la definición de las restricciones de cardinalidad y valor. Las restricciones de valor se utilizan para especificar el tipo de datos del dominio y el rango de una propiedad. Por ejemplo, el rango de la propiedad *fieldIsRequiered* está restringido a booleano. Por el contrario, el dominio de la propiedad *hasRelationFrom* es *Relation*, y el rango está limitado a los individuos de la clase *Node*. El rango de la propiedad *ruleSign* está restringido a los valores "+" o "-". Además, la propiedad del objeto *HasSubRole* se ha marcado como transitiva para que el razonador pueda inferir la jerarquía de roles.

7. *Crear instancias.* Las instancias de la ontología se generan a partir de los metamodelos, que se definen en XMI [118]. Los metamodelos se transforman a RDF mediante funciones de mapeo según las clases y propiedades especificadas en la ontología, creando un grafo de conocimiento en RDF (ver Sección 5.4.2).

5.4.1. Modelo ontológico

Tras aplicar la metodología anterior, se ha desarrollado la ontología. La ontología define un total de 24 clases (que representan individuos con la misma taxonomía), 12 propiedades de objeto

(representan relaciones binarias entre individuos), 7 propiedades de datos (atributos de los individuos) y 112 axiomas de restricción. Así pues, la ontología es lo suficientemente amplia como para permitir razonamientos complejos.

A continuación se enumeran las principales clases acompañadas de una breve descripción:

- La clase ***SecurityRule*** representa las reglas de seguridad en una base de datos genérica. Se han definido las siguientes propiedades de objeto: *SecurityRuleDefined* vincula la regla con el elemento de la base de datos sobre el que está definida, *SecurityRuleContainsPrivilege* vincula la regla con el privilegio que la regla concede o revoca. Hemos definido las subclases *SecurityRuleElement* y *SecurityRuleField* para especificar si se trata de una regla que afecta a un elemento o a un campo en el caso de una base de datos orientada a grafos. *SecurityRuleElement* y *SecurityRuleField* son clases disjuntas, es decir, no existe ninguna regla que pertenezca al mismo tiempo a estas dos Clases.

- La clase ***Privilege*** describe los privilegios asociados a una regla de seguridad. Los privilegios se especializan en 4 subclases: *Create*, *Read*, *Update* y *Delete*.

- La clase ***Role*** modela los roles que se definen en una base de datos. Los roles agrupan un conjunto de reglas que se aplican a los usuarios. La creación de roles facilita la gestión de la seguridad ya que no es necesario asignar reglas individuales a cada usuario. Las propiedades de objeto de la clase *Role* son: *RoleHasRules*; esta propiedad vincula el rol con las reglas y *HasSubRole*, que permite definir jerarquías entre roles.

- La clase ***User*** representa a los usuarios del sistema de gestión de bases de datos. La propiedad del objeto *UserHasRole* vincula a un usuario con sus roles asignados.

- La clase ***Database*** representa el concepto de base de datos de forma genérica. El nombre de la base de datos se almacena con la propiedad de datos *databaseHasName*. *Database* es una subclase de la clase BIGOWL *Data*. *NoSQLDatabases*, en última instancia, *GraphsDatabases* son subclases de *Database*. La propiedad de objeto *hasElement* relaciona una instancia de una base de datos orientada a grafos con los elementos que la componen.

- La clase ***Element*** de acuerdo con la teoría de grafos, la clase *Element* define genéricamente las propiedades comunes de los términos que componen un grafo. La propiedad de datos *elementHasName* almacena el nombre de cada elemento del grafo. La propiedad de objeto *hasElements* relaciona una *GraphDatabase* con sus elementos. Por otro lado, *hasElementField* relaciona individuos de *Element* con individuos de la clase *NoSQLFieldGraph*. *Node* y *Relationship* se especifican como subclases de *Element*; para estas clases, se han incluido dos propiedades de objeto, *hasRelationFrom* y *hasRelationTo*, que asocian los nodos con las relaciones.

- La clase ***Field*** representa el concepto de un campo de datos en cualquier base de datos, independientemente de su implementación. Las propiedades de datos *fieldHasName* y *isRequired* almacenan el nombre del campo y si el campo es requerido; En cuanto a las propiedades de objeto, *hasDatatype* conecta el campo con su tipo de datos. Como subclases, definimos la jerarquía *NoSQLField* y *NOSQLFieldGraph* para anotar el tipo de base de datos específico al que pertenece el campo.

Implementar la ontología en OWL2 permite explotar las capacidades de razonamiento del lenguaje ontológico. La Regla5.1 muestra un ejemplo sencillo de cómo el razonador puede detectar inconsistencias en la definición de reglas de seguridad. Por ejemplo, si una regla de seguridad se define como tipo *Element* y en realidad está definida sobre un Campo, el razonador avisa de la incossistencia.

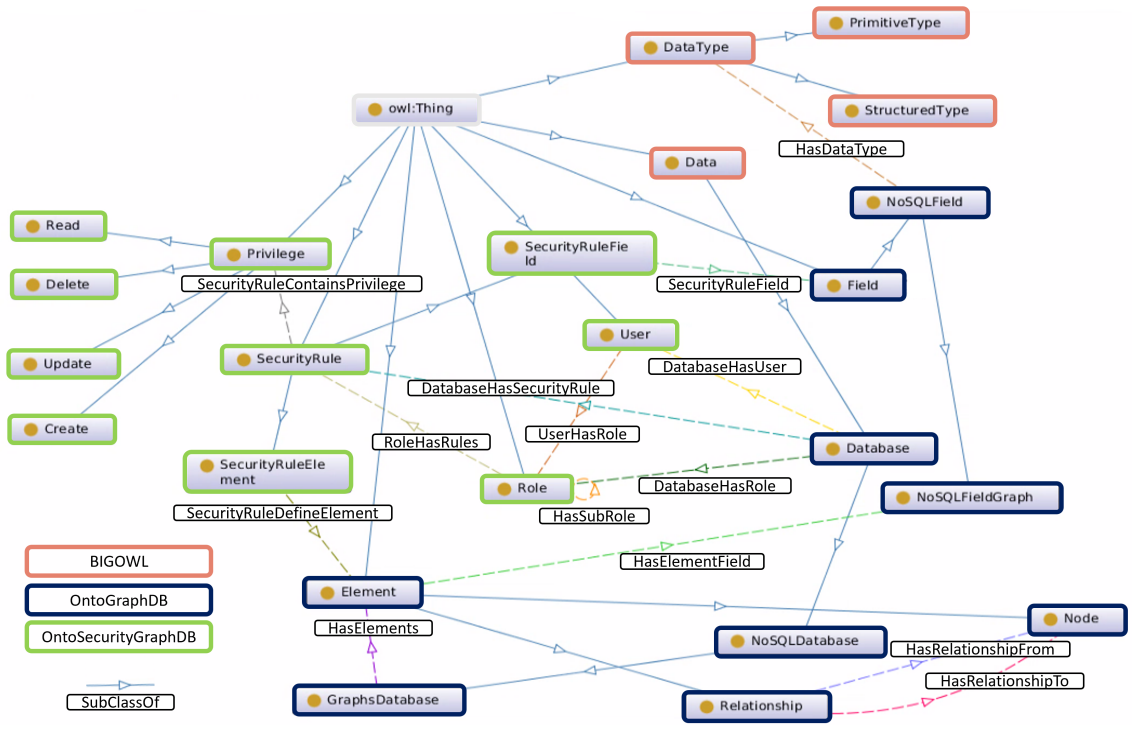

Figura 5.3: Visión general de la ontología de seguridad en bases de datos orientadas a grafos.

Regla 5.1: Ejemplo de regla de razonamiento.

Regla semántica OWL 2:
Sea **C** una clase, **P** una propiedad y dos individuos **x** e **y**.
Si el **Dominio(P,C)** y **P(x,y)** entonces **C(x) (sr1)**

Capa semántica:
Dominio(SecurityRuleDefinedField, SecurityRuleField)
Disjunto Clases(SecurityRuleDefinedField, SecurityRuleDefinedElement) (ax1)

Individuos:
SecurityRuleElement(rule1)
SecurityRuleDefinedField(rule1, field1) → SecurityRuleField(rule1) (por sr1)
Inconsistencia (por ax1)

5.4.2. Consolidación de datos

Tras el desarrollo de la ontología, los metamodelos se traducen a RDF para construir un grafo de conocimiento acorde con la estructura de la ontología. La Figura 5.4 muestra la infraestructura diseñada para dicha transformación.

Se ha desarrollado un conjunto de funciones de mapeo específicas. Estas funciones producen los individuos a partir del fichero XMI siguiendo la jerarquía de clases de la ontología. Las propiedades de objetos y datos de la ontología permiten relacionar los individuos entre sí. Los metamodelos se especifican en XML. La Función 5.2 muestra un extracto del código que crea las tripletas RDF para los privilegios.

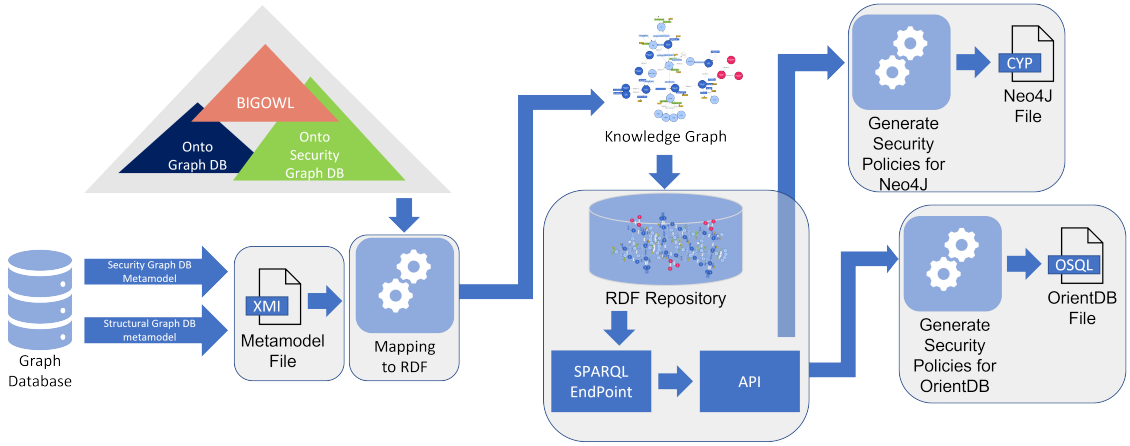

Figura 5.4: Infraestructura para crear el grafo del conocimiento.

Fragmento de código 5.2: Traducción de los privilegios.

```
def parse_privileges(rule, privilege):
    id_privilege = hashlib.sha224(privilege["@xmi:id"])
    uri_privilege = URIRef(sec + id_privilege)

    rdf_graph.add((uri_privilege,RDF.type,sec[privilege["@xsi:type"]]))

    rdf_graph.add((rule,sec.SecurityRuleContainsPrivilege,uri_privilege))

    if "@condition" in privilege:
        rdf_graph.add((uri_privilege, sec.privilegeCondition, Literal(privilege["@condition"])))
```

Una vez creado el grafo de conocimiento, se almacena en un repositorio RDF implementado bajo una instancia de Stardog[2] que cuenta con capacidades de persistencia y razonamiento. En este punto, ya es posible consultar el grafo de conocimiento desde un punto de consulta SPARQL. La Consulta 5.3 muestra un ejemplo de consulta SPARQL sobre el repositorio RDF. Esta consulta recupera los detalles de una regla específica (signo, rol, privilegio, nombre del elemento, nombre de la regla y condición).

También se ha definido una API que envuelve las consultas SPARQL. Esta API tiene como objetivo extraer fácilmente los datos necesarios del grafo de conocimiento para generar las políticas de seguridad para las diferentes implementaciones de bases de datos de grafos. Las llamadas implementadas devuelven los roles, sus hijos directos y sus descendientes, los usuarios, los roles asignados a un usuario, las reglas de seguridad definidas sobre los elementos, las reglas definidas sobre los campos, etc. Al consultar el grafo de conocimiento, obtenemos todos los datos necesarios para crear las políticas de seguridad en una base de datos final orientada a grafos.

Una de las grandes ventajas de desarrollar una ontología es la posibilidad de utilizar capacidades de razonamiento. Por ejemplo, haber declarado la propiedad de objeto *HasSubRole* como transitiva nos permite obtener todos los descendientes de un rol concreto utilizando el razonador al ejecutar la Consulta 5.4; sin utilizar el razonador, la consulta sólo devuelve sus hijos directos. En la Consulta 5.5 el razonador infiere que una base de datos orientada a grafos es una base de datos y devuelve las reglas de seguridad definidas para ella. Sin el razonador, la consulta no obtiene ningún resultado. La ontología fácilmente para incluir otros tipos de bases de datos.

[2]https://www.stardog.com/

Fragmento de código 5.3: Obtener detalles de una regla de seguridad definida sobre elementos.

```
PREFIX sec: <https://w3id.org/OntoSecurityGraphDB/>
PREFIX db: <https://w3id.org/OntoGraphDB/>
SELECT ?sign ?RoleName ?privilege ?elementName ?condition
WHERE{
?role rdf:type sec:Role.
?role sec:roleName ?RoleName.
?role sec:RoleHasRules ?rule.
?rule sec:ruleName ?ruleName. FILTER(?ruleName = "rule_name")
OPTIONAL{
  ?rule sec:ruleSign ?sign.}
OPTIONAL{
  ?rule sec:SecurityRuleContainsPrivilege ?priv_uri.
  ?priv_uri rdfs:label ?privilege.
    OPTIONAL{
      ?priv_uri sec:privilegeCondition ?condition.
    }
}
?rule sec:SecurityRuleDefineElements ?element.
?element db:elementHasName ?elementName.
?element rdf:type ?ty FILTER(?ty!=owl:NamedIndividual).
}
```

Fragmento de código 5.4: Obtener todos los descendientes de un rol.

```
PREFIX sec: <https://w3id.org/OntoSecurityGraphDB/>
SELECT ?RoleName ?subRoleName
WHERE{
?role rdf:type sec:Role.
?role sec:roleName ?RoleName. FILTER(?RoleName = "rol")
    OPTIONAL{
            ?role sec:HasSubRole ?subRole.
            ?subRole sec:roleName ?subRoleName.
    }
}
```

Fragmento de código 5.5: Obtener todas las reglas definidas sobre elementos dado el nombre de la base de datos.

```
PREFIX sec: <https://w3id.org/OntoSecurityGraphDB/>
PREFIX db: <https://w3id.org/OntoGraphDB/>
SELECT ?ruleName
WHERE{
      ?dataBase rdf:type db:Database.
      ?dataBase db:databaseHasName ?DatabaseName. FILTER(?DatabaseName = "db_name")
      ?dataBase sec:DatabaseHasSecurityRule ?rule.
      ?rule rdf:type sec:SecurityRuleElement.
      ?rule sec:ruleName ?ruleName.
}
```

Otra ventaja es que el/la diseñador/a puede modelar la seguridad utilizando conceptos de la ontología de la parte estructural de la base de datos orientada a grafos, como Elemento o Campo,

Figura 5.5: Flujo de trabajo que dado un metamodelo de una base de datos, genera la implementación final en diferentes sistemas de gestión de bases de datos.

sin haber seleccionado aún la tecnología de base de datos orientada a grafos concreta que va a utilizar. Esto permite retrasar la selección si es necesario hasta que se hayan aclarado requisitos como la escalabilidad, la programación del lado del servidor o las propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), entre otros, o hasta que sea necesario implementar políticas de seguridad.

5.5. Implementación

Se ha implementado la propuesta utilizando TITAN como plataforma de orquestación y ejecución de flujos de trabajo. Como se comentó anteriormente, el núcleo de TITAN es la ontología BIGOWL, que nos permite definir diferentes tipos de componentes, sus entradas, salidas y parámetros como tripletas RDF. A través de una interfaz web y una API REST, esta plataforma nos permite componer y ejecutar las instancias del flujo de trabajo. Una de las funcionalidades de TITAN es listar el catálogo de componentes disponibles junto con sus entradas y salidas. La lista de componentes se obtiene mediante llamadas a la API REST que transforma las consultas SPARQL a JSON. Una vez diseñado el flujo de trabajo y validada la compatibilidad entre componentes, se ejecuta el flujo de trabajo. Cada ejecución se anota semánticamente en el repositorio RDF de TITAN como un individuo de tipo tarea. Una tarea es una instancia concreta de un componente con sus parámetros configurados para esa ejecución. Por último, la API REST permite comprobar el estado de ejecución y descargar los archivos generados en cada componente.

El flujo de trabajo se ha implementado mediante 5 componentes interconectados, como muestra la Figura5.5. A continuación, se detalla la funcionalidad de cada componente.

- ***Import file.*** Este componente es el punto de entrada de datos del usuario al flujo de trabajo. Dicho componente tiene como parámetro la URL del archivo que contiene el metamodelo. Como salida, tiene la dirección en el sistema de almacenamiento de ficheros utilizado por TITAN.
- ***Graph model to knowledge graph.*** Este componente traduce los metamodelos a tripletas RDF. Las salidas del componente son un archivo con las tripletas en RDF y un archivo PDF con una representación del grafo de conocimiento.
- ***Insert into knowledge graph.*** Este componente recibe el fichero con las tripletas RDF que contienen la información de los individuos y las ontologías definidas. Las tripletas se almacenan en una base de datos del repositorio RDF. Este componente devuelve como salida el identificador del grafo de conocimiento para que otros componentes puedan utilizarlo.

- ***Knowledge graph to Neo4J*** y ***Knowledge graph to OrientDB***. Estos componentes extraen la información del grafo de conocimiento y la transforman en el lenguaje de definición utilizado por Neo4j y OrientDB para las políticas de seguridad, respectivamente.

En cuanto a la implementación de estos componente, aunque la propuesta descrita en este capítulo es aplicable a cualquier herramienta, nos hemos centrado en Neo4j y OrientDB debido a que Neo4j está considerado como un referente en este tipo de sistemas[3] y OrientDB incluye algunas funcionalidades de seguridad interesantes (que Neo4j no permite), como la representación de políticas de seguridad con condiciones asociadas.

Existen otras herramientas de gestión de bases de datos orientadas a grafos (como JanusGraph, NebulaGraph, Memgraph, TigerGraph, etc.) a las que podríamos aplicar el modelo propuesto, pero no han sido consideradas, por ser menos interesantes desde el punto de vista de la seguridad, ya que suelen presentar sistemas RBAC básicos y ofrecen funcionalidades limitadas para establecer políticas de seguridad.

En primer lugar, se analizan las características que ofrece Neo4J para la definición de políticas de seguridad. La Tabla5.3 muestra la sintaxis concreta en EBNF. Se observa que una política de seguridad incluye información relativa al signo de autorización, privilegios, elementos y roles a los que afecta. En el caso concreto de Neo4j, cabe mencionar que los privilegios clásicos (creación, lectura, modificación y borrado) se especializan en más tipos de privilegios y se agrupan simultáneamente de forma jerárquica. Los permisos de lectura se refieren al acceso al valor (*Read*) o a poder llegar al elemento (*Traverse*), y tiene un privilegio que agrupa los dos anteriores (*Match*). Respecto a la escritura, existe un privilegio que agrupa a los anteriores (*Write*), que a su vez se especializa en la creación (*Create*), el borrado (*Delete*), la modificación (*Set Property*) y la gestión de etiquetas (Set y delete label). Los elementos estructurales también se categorizan en jerarquías, lo que significa que podemos referirnos a nodos (*Node*), relaciones (*Relationship*) o ambos, que se denominan elementos (*Element*). En este sentido, si deseamos establecer una autorización de grano fino para determinadas propiedades, deberemos indicar tanto el nombre de estas propiedades como el de los elementos a los que pertenecen.

Podemos observar, en la Tabla 5.4, cómo sería la sintaxis para la implementación de políticas de seguridad en el otro sistema gestor de bases de datos orientado a grafos, OrientDB.

Su estructura es similar a la sintaxis vista en Neo4j en cuanto a que permite definir políticas de seguridad indicando el signo, privilegios, elementos y roles afectados. Sin embargo, ambas herramientas difieren en el número de políticas que deben definirse para especificar una misma restricción de seguridad. En OrientDB es posible agrupar en una sola política de seguridad varios privilegios (crear, leer, etc.) mientras que en Neo4j es necesario definir una política por privilegio. Por otra parte, OrientDB requiere que cada política afecte a un solo rol, mientras que Neo4j permite especificar una lista de roles. Lo más interesante de OrientDB, a diferencia de Neo4j, es que permite asociar a cada privilegio una condición (*sqlPredicate*) que debe cumplirse. Por ejemplo, de esta forma podríamos representar políticas que permitan a los usuarios consultar o editar sus propios datos. Imaginemos un sistema médico en el que un usuario de rol paciente puede consultar un hipotético nodo paciente (con sus atributos), pero sólo la instancia que hace referencia a sus propios datos.

Una vez analizada la herramienta de destino, desarrollamos un conjunto de transformaciones que, partiendo de la especificación a nivel de diseño, generan automáticamente la correspondiente implementación de las políticas de seguridad.

Estas transformaciones se han implementado como scripts Python empaquetados en un contenedor Docker y hacen uso de la API desarrollada en este trabajo para consultar los recursos semánticos necesarios (estructura de bases de datos orientadas a grafos, modelo de seguridad para bases de datos NoSQL y la información del modelo para una instancia concreta a transformar).

[3] https://db-engines.com/en/ranking/graph+dbms

5.6. Validación

El marco propuesto en este capítulo es aplicable al diseño seguro de cualquier base de datos orientada a grafos. A continuación se presenta un ejemplo de validación en el que la propuesta se aplicada al ámbito sanitario, más concretamente a la gestión de diagnósticos, en los que intervienen pacientes, médicos y tratamientos asociados. A continuación se ilustra cómo un/una diseñador/a aplicaría la propuesta al diseño seguro de esta base de datos, mediante una serie de pasos que podrían extrapolarse al diseño de cualquier otra base de datos orientada a grafos.

En primer lugar, el/la diseñador/a modela la base de datos orientada a grafos a alto nivel. Esto incluye los aspectos estructurales necesarios (Nodos, Relaciones, Campos, etc.) así como las políticas de seguridad necesarias para satisfacer los requisitos de seguridad del sistema.

Esta tarea se realiza sin tener en cuenta características específicas del sistema gestor de base de datos en el que finalmente se implante el sistema, ya que este es un aspecto que nuestra propuesta genera de forma automatizada.

En este ejemplo, las políticas de seguridad del sistema se generan considerando Neo4J como plataforma de destino, comprobando finalmente cómo la implementación generada satisface los requisitos de seguridad establecidos.

Empezando por la parte estructural (Figura 5.6), se definen los siguientes nodos, relaciones y propiedades. El personal de administración (nodo *AdmissionStaff*) se encarga de registrar a los pacientes (nodo *Patient*) y de mantener su información asociada (nombre, dirección, número de la seguridad social, etc.). Los médicos (nodo *Doctor*) tienen una especialidad asociada y se encargan de diagnosticar a los pacientes, de forma que los pacientes tienen asociadas enfermedades (nodo *Disease*) diagnosticadas en una fecha determinada por un médico determinado. Además, cada enfermedad tiene asociados una serie de posibles tratamientos (nodo *Treatment*), siendo el médico el que selecciona uno de ellos como el tratamiento actual que está siguiendo un determinado paciente que padece esa enfermedad.

En cuanto a la parte de seguridad, en primer lugar se decide definir un rol para cada tipo de usuario que podrá interactuar con el sistema: personal de administración (*RoleAdmissionStaff*), pacientes (*RolePatient*) y médicos (*RoleDoctor*). A continuación, para cada uno de estos roles, se establece un conjunto de autorizaciones que limitan sus privilegios de acuerdo a la política de seguridad buscada. Entrando en más detalle, se definen las siguientes reglas de seguridad:

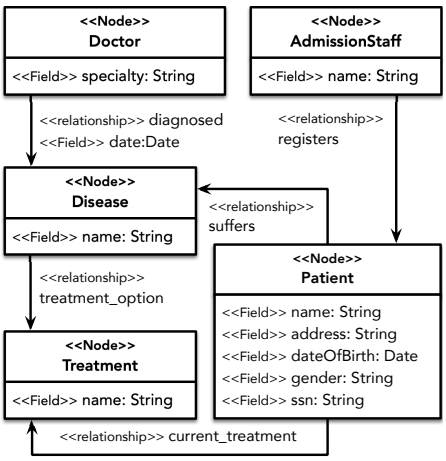

Figura 5.6: Metamodelo de la parte estructural.

- El rol de paciente (Figura 5.7) tiene una autorización positiva definida que permite a los médicos consultar los datos del paciente.
- El rol personal de admisión (Figura 5.8) tiene asociadas varias autorizaciones para otorgar privilegios sobre nodos y relaciones. Por un lado, privilegios de lectura, creación y actualización (no borrado) de pacientes y de la relación que indica que determinado personal ha dado de alta a un determinado paciente. Y por otro, conceder privilegios de lectura sobre el personal de admisión.
- El rol médico (Figura 5.9) presenta dos autorizaciones similares a las vistas anteriormente, que otorgan varios privilegios sobre nodos y relaciones. Pero además, define dos autorizaciones de grano fino sobre propiedades. La primera de ellas establece una autorización negativa que retira el permiso de lectura sobre el número de la seguridad social de los pacientes (sobre el que tenía pleno acceso de lectura). La segunda regla también refina el acceso a la información de los pacientes, esta vez retirando el privilegio de leer sus direcciones, aunque solo para aquellos pacientes menores de edad.

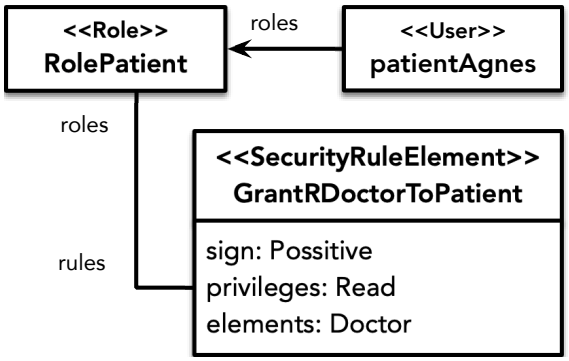

Figura 5.7: Metamodelo de los permisos del rol paciente.

A continuación, ejecutamos el flujo de trabajo en TITAN con los metamodelos de las Figuras 5.6, 5.7, 5.8, 5.9 como entrada. El grafo de conocimiento se crea y almacena en el repositorio RDF. Las Figuras 5.10 y 5.11 muestran parte de dicho grafo de conocimiento, en concreto, los aspectos estructurales de la base de datos y el rol paciente, respectivamente. Los nodos rosa del grafo de conocimiento representan individuos, y los azules, clases de la ontología. Los nodos cuadrados simbolizan valores de datos, mientras que los arcos son propiedades de objetos y datos de la ontología.

Durante la ejecución, los componentes extraen las políticas de seguridad del grafo de conocimiento a través de la API REST, que encapsula las consultas SPARQL. Por ejemplo, con la Consulta 5.5 obtenemos todas las reglas de la base de datos. La Tabla 5.5 muestra el resultado de la ejecución de la consulta. Una vez que tenemos todos los nombres de las reglas, se ejecuta la Consulta 5.3 para recuperar los detalles de las reglas. La Tabla 5.6 expone los detalles de la regla *GrantRNodesToDoctor*. que autoriza al rol *RoleDoctor* a leer varios nodos. Podemos ver como afecta completamente a los nodos Paciente, Enfermedad, Médico y Tratamiento, sin presentar ninguna condición a evaluar (que es un campo opcional de las reglas de seguridad).

El último paso de la propuesta consiste en la generación automatizada de la implementación de las políticas de seguridad definidas en un gestor específico de bases de datos orientadas a grafos. Para este caso de estudio se genera la implementación de las políticas de seguridad especificadas en el modelo para las herramientas Neo4j y OrientDB.

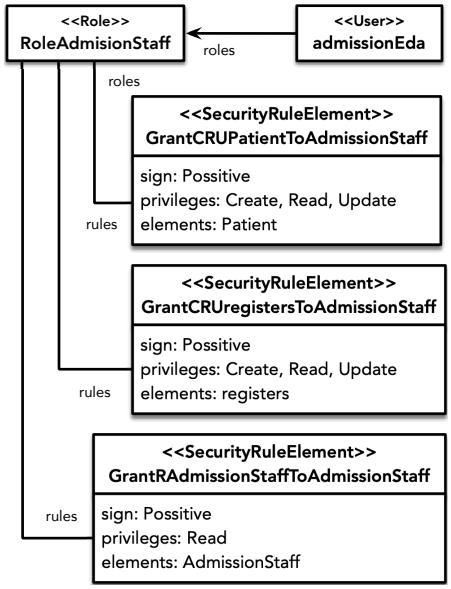

Figura 5.8: Metamodelo de los permisos del rol personal de admisión.

Fragmento de código 5.6: GrantCRUPatientToAdmissionStaff

```
# Neo4j
GRANT CREATE ON GRAPH Hospital NODE Patient TO RoleAdmisionStaff;
GRANT MATCH ON GRAPH Hospital NODE Patient TO RoleAdmisionStaff;
GRANT SET PROPERTY {*} ON GRAPH Hospital NODE Patient TO RoleAdmisionStaff;

# OrientDB
GRANT SET CREATE, READ, AFTER UPDATE ON database.class.Patient TO RoleAdmisionStaff;
```

En los fragmentos de código 5.6 y 5.7 se puede observar a modo de ejemplo cómo se han implementado finalmente varias reglas de seguridad. Por un lado, mostramos la regla "*GrantCRUPatientToAdmissionStaff*", que otorga privilegios de creación, lectura y modificación sobre el elemento Paciente para el rol *RoleAdmissionStaff*. Dicha regla se transforma en Neo4j en varias políticas de seguridad (una por privilegio) y en una única política en OrientDB (Fragmento de código 5.6).

Por otro lado, mostramos la regla "*DenyRPatientaddressToDoctor*" que deniega lecturas sobre el atributo dirección de Paciente al rol Doctor, pero incluye una condición indicando que sólo afectaría a aquellos pacientes menores de 18 años. La definición de estas condiciones no es soportado por Neo4j pero sí por OrientDB, por lo que las transformaciones actuarán siguiendo estrategias diferentes. En Neo4j se opta por una estrategia conservadora en la que se oculta la dirección de todas las instancias del Paciente, mientras que en OrientDB la condición se indica en la política de seguridad (Fragmento de código 5.7).

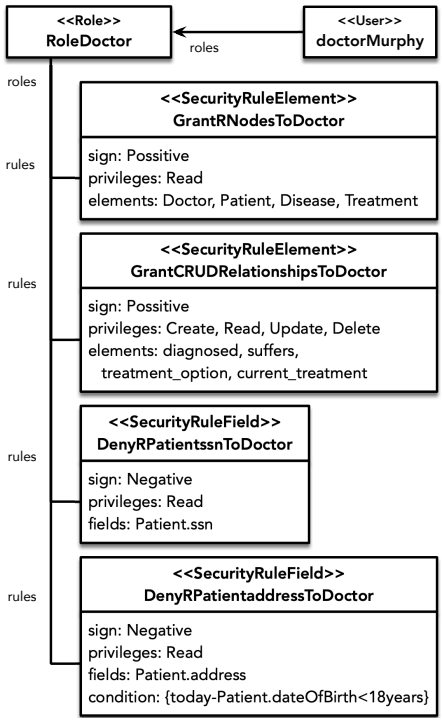

Figura 5.9: Metamodelo permisos del rol médicos.

Fragmento de código 5.7: DenyRPatientaddressToDoctor

```
# Neo4j
DENY MATCH address ON GRAPH Hospital NODE Patient TO RoleDoctor;

# OrientDB
REVOKE SET READ = (today−Patient.dateOfBirth < 18 years) ON database.class.address TO
    RoleDoctor;
```

5.7. Conclusiones

Las bases de datos NoSQL se caracterizan por ofrecer un alto rendimiento y flexibilidad al tiempo que consideran la seguridad como un aspecto secundario. La falta de mecanismos de seguridad estandarizados, incluso entre tecnologías que comparten el mismo tipo de base de datos NoSQL, junto con la ausencia de una metodología segura de alta abstracción, dificulta la creación de diseños seguros. En este contexto, el diseñador de bases de datos tiene que conocer los mecanismos particulares de bajo nivel que ofrece cada tecnología, las limitaciones y las soluciones alternativas para lograr una implementación segura.

Para abordar este problema, en este capítulo se presenta el primer marco de seguridad de alta abstracción para el diseño seguro de bases de datos NoSQL. Uno de los aspectos clave de nuestro enfoque es la propuesta de una ontología, que permite a los diseñadores modelar conceptos y estructuras de alto nivel de las bases de datos NoSQL, junto con las políticas de seguridad requeridas al mismo tiempo y en la misma fase de diseño.

Figura 5.10: Parte del grafo de conocimiento que representa los aspectos estructurales.

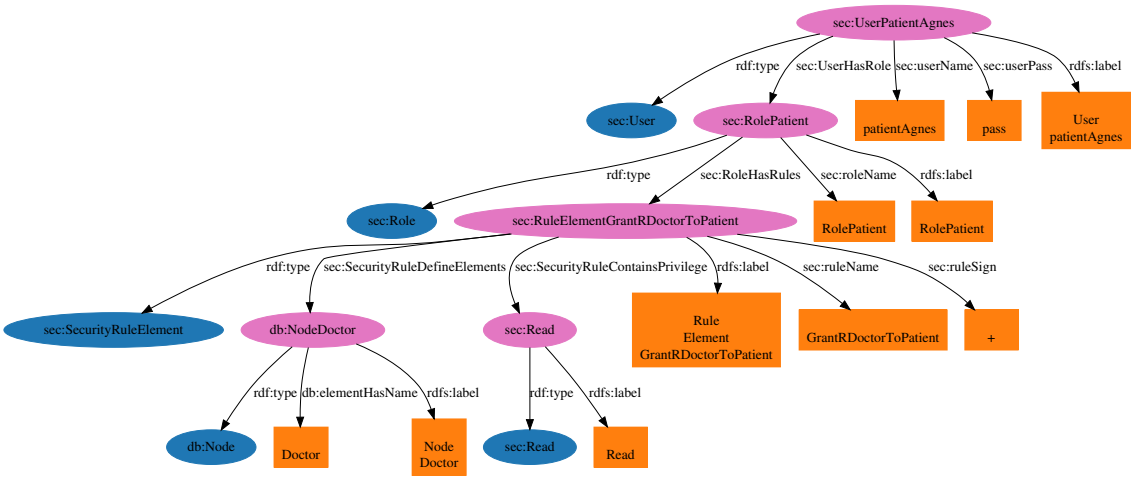

Figura 5.11: Parte del grafo de conocimiento que representa el rol de paciente.

Otra gran ventaja del marco propuesto es que, gracias a la capa ontológica, los diseñadores pueden implementar reglas ontológicas para analizar y detectar posibles errores y problemas pasados por alto. Además, este marco nos permite derivar automáticamente la implementación de las bases de datos NoSQL seguras independientemente de la tecnología concreta que se vaya a utilizar posteriormente, ahorrando tiempo y evitando errores introducidos en implementaciones ad-hoc.

Como trabajo futuro, se estudia ampliar la propuesta a todo tipo de bases de datos. Actualmente, se ha desarrollado y validado la metodología propuesta utilizando una base de datos orientada a grafos como caso de estudio. Sin embargo, se reconoce que existen diferentes tipos de bases de datos con características y estructuras diversas. Por lo tanto, se pretende investigar y adaptar la metodología para su aplicación en una amplia gama de bases de datos, incluyendo bases de datos relacionales, bases de datos NoSQL y bases de datos distribuidas. Esto implicará el estudio de las peculiaridades y desafíos asociados a cada tipo de base de datos, y la adaptación de la metodología para abordar eficazmente sus particularidades. Al ampliar la propuesta a todo tipo de bases de datos, se espera que esta investigación tenga un mayor impacto en el ámbito de la gestión de datos y proporcione soluciones más generales y aplicables a una amplia variedad de entornos y sistemas de bases de datos.

Tabla 5.1: Propiedades de datos y propiedades de objeto de la ontología definida para la seguridad en bases de datos No-SQL.

Propiedades de objeto	Lógica descriptiva
DatabaseHasRole	$\exists$ DatabaseHasRole Thing $\sqsubseteq$ Database $\top \sqsubseteq \forall$ DatabaseHasRole Role
DatabaseHasSecurityRule	$\exists$ DatabaseHasSecurityRule Thing $\sqsubseteq$ Database $\top \sqsubseteq \forall$ DatabaseHasSecurityRule SecurityRule
DatabaseHasUser	$\exists$ DatabaseHasUser Thing $\sqsubseteq$ Database $\top \sqsubseteq \forall$ DatabaseHasUser User
HasSubRole	TransitiveProperty HasSubRole $\exists$ HasSubRole Thing $\sqsubseteq$ Role $\top \sqsubseteq \forall$ HasSubRole Role
RoleHasRules	$\exists$ RoleHasRules Thing $\sqsubseteq$ Role $\top \sqsubseteq \forall$ RoleHasRules SecurityRule
SecurityRuleContainsPrivilege	$\exists$ SecurityRuleContainsPrivilege Thing $\sqsubseteq$ SecurityRule $\top \sqsubseteq \forall$ SecurityRuleContainsPrivilege Privilege $\sqsubseteq$ SecurityRuleDefined
SecurityRuleDefineElements	$\exists$ SecurityRuleDefineElements Thing $\sqsubseteq$ SecurityRuleElement $\top \sqsubseteq \forall$ SecurityRuleDefineElements Element $\sqsubseteq$ SecurityRuleDefine
SecurityRuleDefineField	$\exists$ SecurityRuleDefineField Thing $\sqsubseteq$ SecurityRuleField $\top \sqsubseteq \forall$ SecurityRuleDefineField Field
SecurityRuleDefined	$\exists$ SecurityRuleDefined Thing $\sqsubseteq$ SecurityRule
UserHasRole	$\exists$ UserHasRole Thing $\sqsubseteq$ User $\top \sqsubseteq \forall$ UserHasRole Role
Propiedades de datos	**Lógica descriptiva**
privilegeCondition	$\exists$ privilegeCondition Datatype Literal $\sqsubseteq$ Privilege $\top \sqsubseteq \forall$ privilegeCondition Datatype string
roleName	$\exists$ roleName Datatype Literal $\sqsubseteq$ Role $\top \sqsubseteq \forall$ roleName Datatype string
ruleName	$\exists$ ruleName Datatype Literal $\sqsubseteq$ SecurityRule $\top \sqsubseteq \forall$ ruleName Datatype string
ruleSign	$\exists$ ruleSign Datatype Literal $\sqsubseteq$ SecurityRule $\top \sqsubseteq \forall$ ruleSign {“+”string} $\sqcup$ {”string}
userName	$\exists$ userName Datatype Literal $\sqsubseteq$ User $\top \sqsubseteq \forall$ userName Datatype string
userPass	$\exists$ userPass Datatype Literal $\sqsubseteq$ User $\top \sqsubseteq \forall$ userPass Datatype string

Tabla 5.2: Propiedades de datos y objeto de la ontología definida para bases de datos orientadas a grafos.

Propiedades de objeto	Lógica descriptiva
hasDatatype	$\exists$ hasDatatype Thing $\sqsubseteq$ NoSQLField $\top \sqsubseteq \forall$ hasDatatype DataType
hasElementField	$\exists$ hasElementField Thing $\sqsubseteq$ Element $\top \sqsubseteq \forall$ hasElementField NoSQLFieldGraph
hasElements	$\exists$ hasElements Thing $\sqsubseteq$ GraphsDatabase $\top \sqsubseteq \forall$ hasElements Element
hasRelationFrom	$\exists$ hasRelationFrom Thing $\sqsubseteq$ Relation $\top \sqsubseteq \forall$ hasRelationFrom Node
hasRelationTo	$\exists$ hasRelationTo Thing $\sqsubseteq$ Relation $\top \sqsubseteq \forall$ hasRelationTo Node
Propiedades de datos	**Lógica descriptiva**
databaseHasName	$\exists$ databaseHasName Datatype Literal $\sqsubseteq$ Database $\top \sqsubseteq \forall$ databaseHasName Datatype string
elementHasName	$\exists$ elementHasName Datatype Literal $\sqsubseteq$ Element $\top \sqsubseteq \forall$ elementHasName Datatype string
fieldIsIndex	$\exists$ fieldIsIndex Datatype Literal $\sqsubseteq$ NoSQLFieldGraph $\top \sqsubseteq \forall$ fieldIsIndex Datatype boolean
fieldHasName	$\exists$ fieldHasName Datatype Literal $\sqsubseteq$ NoSQLField $\top \sqsubseteq \forall$ fieldHasName Datatype string
fieldIsRequired	$\exists$ fieldIsRequired Datatype Literal $\sqsubseteq$ NoSQLField $\top \sqsubseteq \forall$ fieldIsRequired Datatype boolean

Tabla 5.3: Sintaxis para especificar políticas de seguridad en Neo4j.

$< securitypolicy >::=< sign >< privilege >$ $[< properties >]ON < graph >< entity > TO < role >$
$< sign >::= GRANT \mid DENY$
$< privilege >::= MATCH \mid TRAVERSE \mid READ \mid$ $WRITE \mid CREATE \mid DELETE \mid SETPROPERTY \mid$ $SETLABEL \mid DELETELABEL$
$< properties >::= "\{" * \mid < nameList > "\}"$
$< graph >::= DEFAULTGRAPH \mid$ $(GRAPH[S](* \mid < nameList >))$
$< entity >::=< entityType > (* \mid < nameList >)$
$< entityType >::= ELEMENT[S] \mid NODE[S] \mid$ $RELATIONSHIP[S]$
$< roles >::=< nameList >$
$< nameList >::=< name > [", " < name >]*$

Tabla 5.4: Sintaxis para especificar políticas de seguridad en OrientDB.

$< securitypolicy >::=< sign >< listOfPrivileges >$ $ON < element > TO < role >$
$< sign >::= GRANT \mid REVOKE$
$< privilege >::=< action > [= (< sqlPredicate >)]?,$
$< action >::= CREATE \mid READ \mid BEFOREUPDATE \mid$ $AFTERUPDATE \mid DELETE \mid EXECUTE$
$< element >::= database.class.[* \mid < name >]$

Tabla 5.5: Resultados de la Consulta 5.5.

Nombre de la regla
GrantCRUregistersToAdmissionStaff
GrantCRUPatientToAdmissionStaff
GrantRNodesToDoctor
GrantRAdmissionStaffToAdmissionStaff
GrantCRUDRelationshipsToDoctor
GrantRDoctorToPatient

Tabla 5.6: Resultados de la Consulta 5.3.

Signo	Nombre del Rol	Privilegio	Nombre del elemento	Condición
+	RoleDoctor	Read	Patient	
+	RoleDoctor	Read	Disease	
+	RoleDoctor	Read	Doctor	
+	RoleDoctor	Read	Treatment	

Parte III
Observaciones finales

En este Capítulo final, presentamos las conclusiones en la Sección 6.1, donde se destaca el impacto de esta Tesis en la definición semántica de los datos del dominio, orientada a mejorar la reutilización y la integración de datos, así como la importancia de la seguridad en el paradigma de grandes cantidades de datos interconectados. Finalmente, en la Sección 6.2, se plantea una visión hacia adelante que identifica áreas de mejora y futuras investigaciones para perfeccionar la implementación de estas tecnologías en entornos distribuidos y enlazados.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

La presente Tesis Doctoral aborda a lo largo de sus capítulos un enfoque semántico innovador y aplicado, para la integración y el análisis de datos en diferentes áreas de conocimiento. Se obtiene pues una serie de conclusiones específicas de cada dominio de aplicación, las cuales nos llevan a su vez a extraer conclusiones generales y orientadas a responder aquellas preguntas de investigación planteadas en la introducción.

Como principales contribuciones específicas se extraen las siguientes:

- En el Capítulo 3 se demuestra que el enfoque semántico propuesto integra adecuadamente los datos de sistemas de gestión de "e-Learning", permitiendo la consulta avanzada y constituyendo un grafo de conocimiento bien fundamentado para mejorar el análisis informativo en el contexto de los LMSs. Esto lleva a la ontología e-LION propuesta a proporcionar un valor científico añadido, que en el contexto del estado del arte actual (como se explica en la Sección 3.2), permite la conexión semántica con otras ontologías y vocabularios relacionados, promoviendo así la generación de amplios datos enlazados en el dominio del aprendizaje "online". Por lo tanto, la propuesta puede utilizarse en el núcleo de una estrategia de consolidación de datos para futuras aplicaciones, en las que los LMS actuales y otras fuentes de datos académicos se consulten sistemáticamente para apoyar a los profesores con análisis y visualizaciones avanzadas de lo que está ocurriendo en sus asignaturas. Del mismo modo, permite analizar las actividades de los alumnos en el contexto del rendimiento global en un determinado curso, lo que les proporcionaría una perspectiva global de su rendimiento en el curso, fomentando así su aprendizaje proactivo.

- En el contexto de la banca abierta (Capítulo 4), se introduce la Open Bank Ontology (OBO) que define un modelo semántico para la consolidación y anotación de datos de transacciones bancarias según el paradigma que define la normativa PSD2. Este modelo permite consultas SPARQL y razonamiento SWRL, facilitando la reconciliación de datos y la clasificación de clientes según patrones de deuda. Esta ontología fomenta extensiones y federaciones con grafos de conocimiento relacionados en el dominio Fintech. Cabe destacar que la ontología propuesta constituye el primer intento de modelar los movimientos y actividades bancarias con especial atención al estándar europeo PDS2. Por lo tanto, se esperan nuevas extensiones de este trabajo, incluyendo la federación con otros grafos de conocimiento relacionados en el dominio de las aplicaciones y estándares de las Fintech.

- Igualmente, en el Capítulo 5 se aborda la seguridad en bases de datos NoSQL con un marco de alta abstracción enfocado en un diseño seguro. Este marco, que incluye una ontología para modelar conceptos y políticas de seguridad, brinda una metodología esencial para diseñar bases de datos NoSQL seguras, independientemente de la tecnología específica. Además, la capa ontológica permite análisis y detección de errores en la fase de diseño, optimizando la implementación segura.

Como conclusiones generales, a lo largo de esta memoria de Tesis Doctoral se muestra cómo los modelos semánticos basados en ontologías, junto con los grafos de conocimiento, son herramientas poderosas y flexibles para transformar datos dispersos y heterogéneos en conocimiento homogéneo, estructurado y relacionado. Se da soporte así a la la integración, armonización y estandarización de datos, respondiendo a su vez a la primera pregunta de investigación (Q1) expuesta en la introducción.

No obstante, se hace necesaria la definición precisa de "mapeos de datos" desde las fuentes originales hacia un modelo de representación unificado, obteniendo como resultado un grafo de conocimiento definido formalmente por la ontología de dominio específica. Este mapeo proporciona la clave para traducir la diversidad de formatos, estructuras y semántica de los datos de origen a un formato estandarizado, facilitando así su integración y posterior análisis. De hecho, la elaboración de estos "mapeos" requiere un profundo entendimiento del dominio específico y de las características intrínsecas de cada fuente de datos. Es necesario definir reglas y transformaciones que permitan convertir la información desde su formato original a un formato comprensible y coherente en el grafo de conocimiento. Este proceso de mapeo no sólo implica la traducción de esquemas y tipos de datos, sino también la asignación semántica adecuada, donde conceptos de las fuentes de datos se relacionan de manera precisa con conceptos en el grafo de conocimiento. Se automatiza por tanto la conciliación de datos y su organización de cara a la explotación en diferentes dominios, como los abordados en este trabajo, e-Learning, banca abierta y seguridad, dando respuesta así a las preguntas de investigación Q2, Q3 y Q4, respectivamente.

Como aporte adicional, las ontologías propuestas se han diseñado e implementado siguiendo los principios FAIR; también se han definido funciones de mapeo de los datos originales para cada dominio tal y como planteamos en los retos de investigación.

En definitiva, estas conclusiones reflejan avances significativos en la integración semántica de datos y su aplicación en contextos diversos, señalando la utilidad y relevancia de los modelos propuestos. Además, establecen bases sólidas para futuras investigaciones y extensiones en cada dominio, contribuyendo al progreso continuo en la integración y análisis avanzado de los datos.

6.2. Trabajo futuro

Como futuras líneas de investigación en general, nos planteamos continuar con esta propuesta de integración y análisis de datos heterogéneos para mejorar el acceso y su conexión con técnicas de análisis y de explicabilidad. El objetivo principal es utilizar la capa semántica como soporte a la interpretación de análisis de cara al usuario experto del dominio, mejorando así tanto la calidad de los resultados, como la transparencia en el empleo de los algoritmos de análisis.

Otro aspecto importante para considerar en el futuro cercano consiste en diseñar nuevas ontologías y estrategias de datos basadas en tecnologías de la Web Semántica, de soporte a la generación de "Espacios de Datos" (en inglés "Data Spaces"). Estos espacios se están articulando como piezas clave para la estandarización y la regulación de la nueva economía del dato, tanto en entornos industriales y académicos, como administrativos y organizacionales. En este sentido, ya se están

llevando a cabo iniciativas como "GAIA-X"[1] y "European Data Spaces"[2], de gran relevancia a nivel nacional y europeo, en las que se considera la capa semántica y el enlazado de datos como requisito fundamental en estos espacios.

Siguiendo con esta línea, relación simbiótica entre los bots conversacionales basados en grandes modelos lingüísticos (LLM), como ChatGPT, y el concepto de web semántica (pública, privada o híbrida)[1]. Esta simbiosis se debe a las razones expuestas anteriormente en la respuesta de ChatGPT, que pueden resumirse como sigue:

Los LLM proporcionan una base para un potente procesamiento del lenguaje natural basado en su comprensión de la sintaxis y la semántica de las frases; por ejemplo, comprenden la semántica subyacente de múltiples variaciones de la misma frase. Una web semántica es simplemente una variación de la web construida explícitamente a partir de frases que utilizan hipervínculos para expresar la sintaxis y la semántica de la frase de forma computable por la máquina; el efecto neto es un colectivo ilimitado de datos estructurados que manifiesta un grafo global de relaciones entre entidades (en lugar de una red), que comprende una semántica de tipo relación entre entidades computable por la máquina.

Además, considerando diferentes aspectos de las áreas de conocimiento relacionadas con las aportaciones propuestas en estos estudios, se han identificado diferentes líneas de investigación para futuros trabajos. En este apartado se presentan las más destacadas:

- En el contexto del *"e-Learning"*, se plantea incluir más datos de otros sistemas de gestión del aprendizaje *"online"*, así como actualizar la ontología e-LION para incorporar nuevos atributos relevantes desde diferentes perspectivas de los LMSs. En este sentido, otra actividad futura es la alineación ontológica de muchas otras no sólo en el dominio del conocimiento educativo, sino también en diferentes dominios, tales como: redes sociales, comportamientos de los usuarios de Covid-19 relacionados con la salud, evolución demográfica y social.

- En el dominio Fintech, se preveen integrar más datos de diferentes sistemas de gestión de facturas y actualizar la ontología OBO para incorporar nuevos atributos relevantes desde diferentes perspectivas, tales como: opiniones en redes sociales, rasgos de comportamiento de la empresa en sus relaciones comerciales con los clientes, etc. Estos nuevos conocimientos permitirán realizar nuevos análisis, teniendo en cuenta más factores y actores.

- Por último, dentro del campo de la seguridad, se pretende extender la propuesta de esta Tesis a todo tipo de bases de datos. Actualmente, se ha desarrollado y validado la metodología propuesta utilizando una base de datos orientada a grafos como caso de estudio. Sin embargo, se reconoce que existen diferentes tipos de bases de datos con características y estructuras diversas. Por lo tanto, se pretende investigar y adaptar la metodología para su aplicación en una amplia gama de bases de datos, incluyendo bases de datos relacionales, bases de datos NoSQL y bases de datos distribuidas. Esto implicará el estudio de las peculiaridades y desafíos asociados a cada tipo de base de datos, y la adaptación de la metodología para abordar eficazmente sus particularidades. Al ampliar la propuesta a todo tipo de bases de datos, se espera que esta investigación tenga un mayor impacto en el ámbito de la gestión de datos y proporcione soluciones más generales y aplicables a una amplia variedad de entornos y sistemas de bases de datos.

[1]Sitio web de GAIA-X (última visita 16-10-2023) `https://gaia-x.eu/`

[2]Sitio web de Common European Data Spaces (última visita 16-10-2023) `https://dataspaces.info/common-european-data-spaces/`

Índice de Tablas

Índice de figuras

Bibliografía

[1] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood y H. M. Abbasi. «A survey of ontology learning techniques and applications». *Database: The Journal of Biological Databases and Curation* 2018 (2018).

[2] K. Kellou-Menouer, N. Kardoulakis, G. Troullinou, Z. Kedad, D. Plexousakis y H. Kondylakis. «A survey on semantic schema discovery». *The VLDB Journal* 31 (jul. de 2022). DOI: 10.1007/s00778-021-00717-x.

[3] B. Sharma. «Web Semantics and Knowledge Graph». Abr. de 2021, págs. 89-107. ISBN: 978-981-33-6517-9. DOI: 10.1007/978-981-33-6518-6_5.

[4] A. Benítez-Hidalgo, C. Barba-González, J. García-Nieto, P. Gutiérrez-Moncayo, M. Paneque, A. J. Nebro, M. del Mar Roldán-García, J. F. Aldana-Montes e I. Navas-Delgado. «TITAN: A knowledge-based platform for Big Data workflow management». *Knowledge-Based Systems* 232 (2021), pág. 107489.

[5] M. Kulmanov, F. Z. Smaili, X. Gao y R. Hoehndorf. «Semantic similarity and machine learning with ontologies». *Briefings in Bioinformatics* 22 (2020).

[6] Pastorcito. *Arquitectura Tecnológica de la Web Semántica*. [Online; accessed 12-July-2023]. 2012. URL: https://commons.wikimedia.org/wiki/File:Arquitectura_Tecnol%C3%B3gica_de_la_Web_Sem%C3%A1ntica.png#file.

[7] N. F. Noy, D. L. McGuinness et al. *Ontology development 101: A guide to creating your first ontology*. 2001.

[8] N. Guarino et al. «Formal ontology and information systems». *Proceedings of FOIS*. Vol. 98. 1998, págs. 81-97.

[9] B. McBride. «The resource description framework (RDF) and its vocabulary description language RDFS». *Handbook on ontologies*. Springer, 2004, págs. 51-65.

[10] S. Staab y R. Studer. *Handbook on ontologies*. Springer Science & Business Media, 2013.

[11] T. R. Gruber et al. «A translation approach to portable ontology specifications». *Knowledge acquisition* 5.2 (1993), págs. 199-220.

[12] D. L. McGuinness, F. Van Harmelen et al. «OWL web ontology language overview». *W3C recommendation* 10.10 (2004), pág. 2004.

[13] W. O. W. Group. *OWL 2 Web Ontology Language: Document Overview*. http://www.w3.org/TR/owl2-overview/. [Online; accessed 5-April-2018]. 2019.

[14] S. Harris, A. Seaborne y E. Prud'hommeaux. «SPARQL 1.1 query language». *W3C recommendation* 21.10 (2013).

[15] E. Prud, A. Seaborne et al. «SPARQL query language for RDF». *W3C recommendation* (2006).

[16] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof y M. Dean. «SWRL: A Semantic Web Rule Language Combining OWL and RuleML». *W3C Member submission* (mayo de 2004).

[17] B. Magnini, M. Negri, E. Pianta, L. Romano, M. Speranza, L. Serafini, C. Girardi, V. Bartalesi y R. Sprugnoli. «From Text to Knowledge for the Semantic Web: the ONTOTEXT Project». *SWAP 2005 - Semantic Web Applications and Perspectives, Proceedings of the 2nd Italian Semantic Web Workshop, University of Trento, Trento, Italy, 14-16 December 2005*. Ed. por P. Bouquet y G. Tummarello. Vol. 166. CEUR Workshop Proceedings. CEUR-WS.org, 2005. URL: https://ceur-ws.org/Vol-166/36.pdf.

[18] L. Ehrlinger y W. Wöß. «Towards a definition of knowledge graphs.» *SEMANTiCS (Posters, Demos, SuCCESS)* 48.1-4 (2016), pág. 2.

[19] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur e Y. Katz. «Pellet: A practical OWL-DL reasoner». *J. Web Semant.* 5 (2007), págs. 51-53.

[20] T. Berners-Lee. «Linked Data» (jun. de 2006). URL: http://www.w3.org/DesignIssues/LinkedData.html.

[21] M. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. O. Bonino da Silva Santos, P. Bourne, J. Bouwman, A. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. Evelo, R. Finkers y B. Mons. «The FAIR Guiding Principles for scientific data management and stewardship». *Scientific Data* 3 (mar. de 2016). DOI: 10.1038/sdata.2016.18.

[22] S. Russell, P. Norvig y R. Gutiérrez. *Inteligencia artificial: un enfoque moderno*. Colección de Inteligencia Artificial. Pearson Educación, 1996. ISBN: 9789688806821. URL: https://books.google.es/books?id=Z-d9PAAACAAJ.

[23] J. Kuzilek, M. Hlosta y Z. Zdrahal. «Open University Learning Analytics dataset». *Scientific Data* 4 (2017). ISSN: 2052-4463. DOI: https://doi.org/10.1038/sdata.2017.171.

[24] D. Dessì, G. Fenu, M. Marras y D. Reforgiato Recupero. «COCO: Semantic-Enriched Collection of Online Courses at Scale with Experimental Use Cases». *Trends and Advances in Information Systems and Technologies*. Ed. por Á. Rocha, H. Adeli, L. P. Reis y S. Costanzo. Cham: Springer International Publishing, 2018, págs. 1386-1396. DOI: https://doi.org/10.1007/978-3-319-77712-2_133.

[25] S. Karasu, A. Altan, Z. Saraç y R. Hacioğlu. «Prediction of Bitcoin prices with machine learning methods using time series data». *2018 26th Signal Processing and Communications Applications Conference (SIU)*. 2018, págs. 1-4. DOI: 10.1109/SIU.2018.8404760.

[26] S. Karasu y A. Altan. «Recognition Model for Solar Radiation Time Series based on Random Forest with Feature Selection Approach». *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. 2019, págs. 8-11. DOI: 10.23919/ELECO47770.2019.8990664.

[27] A. Altan y S. Karasu. «THE EFFECT OF KERNEL VALUES IN SUPPORT VECTOR MACHINE TO FORECASTING PERFORMANCE OF FINANCIAL TIME SERIES». *The Journal of Cognitive Systems* 4.1 (2019), págs. 17 -21. ISSN: 2548-0650.

[28] M. del Mar Roldán García, J. García-Nieto y J. F. Aldana-Montes. «An ontology-based data integration approach for web analytics in e-commerce». *Expert Systems with Applications* 63 (2016), págs. 20-34. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2016.06.034.

[29] S. Thaddeus, A. Jeganathan y G. T. Leema. «Semantic Integration of Classical and Digital Libraries». *Multimedia Information Extraction and Digital Heritage Preservation*. 2011, págs. 51-65. DOI: 10.1142/9789814307260_0003.

[30] T. Sobral, T. Galvão y J. Borges. «An Ontology-based approach to Knowledge-assisted Integration and Visualization of Urban Mobility Data». *Expert Systems with Applications* 150 (2020), pág. 113260. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113260. URL: https://www.sciencedirect.com/science/article/pii/S0957417420300853.

[31] M. del Mar Roldán-García, S. Uskudarli, N. B. Marvasti, B. Acar y J. F. Aldana-Montes. «Towards an ontology-driven clinical experience sharing ecosystem: Demonstration with liver cases». *Expert Systems with Applications* 101 (2018), págs. 176-195. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.02.001.

[32] M. Brochhausen, J. M. Whorton, C. E. Zayas, M. P. Kimbrell, S. J. Bost, N. Singh, C. Brochhausen, K. W. Sexton y B. Blobel. «Assessing the Need for Semantic Data Integration for Surgical Biobanks—A Knowledge Representation Perspective». *Journal of Personalized Medicine* 12.5 (2022). ISSN: 2075-4426. DOI: 10.3390/jpm12050757. URL: https://www.mdpi.com/2075-4426/12/5/757.

[33] K. McGlinn, M. A. Rutherford, K. Gisslander, L. Hederman, M. A. Little y D. O'Sullivan. «FAIRVASC: A semantic web approach to rare disease registry integration». *Computers in Biology and Medicine* 145 (2022), pág. 105313. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2022.105313. URL: https://www.sciencedirect.com/science/article/pii/S0010482522001056.

[34] J. F. Aldana-Martín, J. García-Nieto, M. del Mar Roldán-García y J. F. Aldana-Montes. «Semantic modelling of Earth Observation remote sensing». *Expert Systems with Applications* 187 (2022), pág. 115838. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.115838. URL: https://www.sciencedirect.com/science/article/pii/S0957417421012008.

[35] P. Delgoshaei, M. Heidarinejad y M. A. Austin. «A Semantic Approach for Building System Operations: Knowledge Representation and Reasoning». *Sustainability* 14.10 (2022). ISSN: 2071-1050. DOI: 10.3390/su14105810. URL: https://www.mdpi.com/2071-1050/14/10/5810.

[36] N. W. Rahayu, R. Ferdiana y S. S. Kusumawardani. «A systematic review of ontology use in E-Learning recommender system». *Computers and Education: Artificial Intelligence* 3 (2022), pág. 100047. ISSN: 2666-920X. DOI: https://doi.org/10.1016/j.caeai.2022.100047. URL: https://www.sciencedirect.com/science/article/pii/S2666920X22000029.

[37] S. R. Heiyanthuduwage. «A Review: Status Quo and Current Trends in E-Learning Ontologies». *Mobility for Smart Cities and Regional Development - Challenges for Higher Education*. Ed. por M. E. Auer, H. Hortsch, O. Michler y T. Köhler. Cham: Springer International Publishing, 2022, págs. 114-125. ISBN: 978-3-030-93904-5.

[38] S. Suguna, V. Sundaravadivelu y B. Gomathi. «A novel semantic approach in E-learning information retrieval system». *2016 IEEE International Conference on Engineering and Technology (ICETECH)*. 2016, págs. 884-889. DOI: 10.1109/ICETECH.2016.7569374.

[39] B. Hssina, B. Bouikhalene y A. Merbouha. «An Ontology to Assess the Performances of Learners in an e-Learning Platform Based on Semantic Web Technology: Moodle Case Study». *Europe and MENA Cooperation Advances in Information and Communication Technologies*. Ed. por Á. Rocha, M. Serrhini y C. Felgueiras. Cham: Springer International Publishing, 2017, págs. 103-112. ISBN: 978-3-319-46568-5.

[40] J. K. Tarus, Z. Niu y A. Yousif. «A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining». *Future Generation Computer Systems* 72 (2017), págs. 37 -48. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.02.049.

[41] K. Makwana, J. Patel y P. Shah. «An Ontology Based Recommender System to Mitigate the Cold Start Problem in Personalized Web Search». *Inf. and Comm. Tech. for Intel. Sys.* Ed. por S. C. Satapathy y A. Joshi. Vol. 1. Cham: Springer, 2018, págs. 120-127.

[42] S. Ouf, M. Abd Ellatif, S. Salama e Y. Helmy. «A proposed paradigm for smart learning environment based on semantic web». *Computers in Human Behavior* 72 (2017), págs. 796 -818. ISSN: 0747-5632.

[43] C. Obeid, I. Lahoud, H. El Khoury y P.-A. Champin. «Ontology-Based Recommender System in Higher Education». *Companion Proceedings of the The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences, 2018, 1031–1034. ISBN: 9781450356404.

[44] L. Han. «An Interdisciplinary Intelligent Teaching System Model Based on College Students' Cognitive Ability». *2018 Int. Conf. on Virtual Reality and Intel. Sys. (ICVRIS)*. 2018, págs. 259-262.

[45] B. Bouihi y M. Bahaj. «An UML to OWL based approach for extracting Moodle's Ontology for Social Network Analysis». *Procedia Computer Science* 148 (2019), págs. 313 -322. ISSN: 1877-0509.

[46] J. Joy, N. S. Raj y R. V. G. «Ontology-Based E-Learning Content Recommender System for Addressing the Pure Cold-Start Problem». *J. Data and Information Quality* 13.3 (abr. de 2021). ISSN: 1936-1955. DOI: 10.1145/3429251. URL: https://doi.org/10.1145/3429251.

[47] M. Al-yahya, R. P. George y A. A. Alfaries. «Ontologies in E-Learning: Review of the literature». *International Journal of Software Engineering and its Applications* 9.2 (2015), págs. 67-84.

[48] C. K. Pereira, S. W. M. Siqueira, B. P. Nunes y S. Dietze. «Linked Data in Education: A Survey and a Synthesis of Actual Research and Future Challenges». *IEEE Transactions on Learning Technologies* 11.3 (2018), págs. 400-412. DOI: 10.1109/TLT.2017.2787659.

[49] S. K., P. Poscic y D. Jaksic. «Ontologies in education – state of the art». *Education and Information Technologies* 25 (2020), 5301–5320. DOI: https://doi.org/10.1007/s10639-020-10226-z.

[50] G. George y A. M. Lal. «Review of ontology-based recommender systems in e-learning». *Computers & Education* 142 (2019), pág. 103642. ISSN: 0360-1315. DOI: https://doi.org/10.1016/j.compedu.2019.103642.

[51] N. F. Noy y D. L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Inf. téc. Stanford University Knowledge Systems Laboratory KSL-01-05, 2001. URL: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

[52] C. Barba-González, J. García-Nieto, M. del Mar Roldán-García, I. Navas-Delgado, A. J. Nebro y J. F. Aldana-Montes. «BIGOWL: Knowledge centered Big Data analytics». *Expert Systems with Applications* 115 (2019), págs. 543-556. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.08.026. URL: https://www.sciencedirect.com/science/article/pii/S0957417418305347.

[53] A. B. Firdausiah Mansur y N. Yusof. «Social learning network analysis model to identify learning patterns using ontology clustering techniques and meaningful learning». *Comps. & Edu.* 63 (2013), págs. 73 -86. ISSN: 0360-1315.

[54] Q. Zeng, Z. Zhao e Y. Liang. «Course ontology-based user's knowledge requirement acquisition from behaviors within e-learning systems». *Computers & Education* 53.3 (2009), págs. 809 -818. ISSN: 0360-1315.

[55] R. Navarrete y S. Luján-Mora. «Use of Linked Data to enhance Open Educational Resources». *2015 International Conference on Information Technology Based Higher Education and Training (ITHET)*. 2015, págs. 1-6. DOI: 10.1109/ITHET.2015.7218017.

[56] M. Polasik, A. Huterska, R. Iftikhar y Štěpán Mikula. «The impact of Payment Services Directive 2 on the PayTech sector development in Europe». *Journal of Economic Behavior & Organization* 178 (2020), págs. 385-401. ISSN: 0167-2681. DOI: https://doi.org/10.1016/j.jebo.2020.07.010.

[57] T. F. Gordon. «An Overview of the Legal Knowledge Interchange Format». *Business Information Systems Workshops*. Ed. por W. Abramowicz, R. Tolksdorf y K. Węcel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, págs. 240-242. ISBN: 978-3-642-15402-7.

[58] P. Castells, B. Foncillas, R. Lara, M. Rico y J. L. Alonso. «Semantic Web Technologies for Economic and Financial Information Management». *The Semantic Web: Research and Applications*. Ed. por C. J. Bussler, J. Davies, D. Fensel y R. Studer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, págs. 473-487.

[59] M. Bennett. «The financial industry business ontology: Best practice for big data». *Journal of Banking Regulation* 14 (jul. de 2013). DOI: 10.1057/jbr.2013.13.

[60] H. Tang y L. Song. «Ontologies in financial services: Design and applications». *International Conference on Business Management and Electronic Information*. Vol. 5. 2011, págs. 364-367. DOI: 10.1109/ICBMEI.2011.5914496.

[61] R. Lara, I. Cantador y P. Castells. «Semantic Web Technologies For The Financial Domain». *The Semantic Web: Real-World Applications from Industry*. Ed. por J. Cardoso, M. Hepp y M. D. Lytras. Boston, MA: Springer US, 2008, págs. 41-74. ISBN: 978-0-387-48531-7. DOI: 10.1007/978-0-387-48531-7_3. URL: https://doi.org/10.1007/978-0-387-48531-7_3.

[62] B. Bhasuran, G. Murugesan, S. Abdulkadhar y J. Natarajan. «Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases». *Journal of Biomedical Informatics* 64 (2016), págs. 1-9. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2016.09.009.

[63] N. Sehgal y A. Crampton. «Information Extraction for Additive Manufacturing Using News Data». Mayo de 2019, págs. 132-138. ISBN: 978-3-030-20947-6. DOI: 10.1007/978-3-030-20948-3_12.

[64] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider y S. Rudolph. «OWL 2 Web Ontology Language Primer (Second Edition)». 2012.

[65] C. Fischer-Pauzenberger y W. Schwaiger. «The OntoREA Accounting Model: Ontology-based Modeling of the Accounting Domain». *Complex Systems Informatics and Modeling Quarterly* (jul. de 2017), págs. 20-37. DOI: 10.7250/csimq.2017-11.02.

[66] G. Guizzardi, G. Wagner, J. P. Almeida y R. S. Andradea Guizzardi. «Towards Ontological Foundations for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story». *Applied Ontology* 10 (2015), págs. 259-271. DOI: https://doi.org/10.1016/j.eswa.2018.02.001.

[67] I. Blums y H. H. Weigand. «Towards a Core Ontology for Financial Reporting Information Systems (COFRIS)». *OTM Workshops*. 2017.

[68] N. Noy. «Ontology Development 101: A Guide to Creating Your First Ontology». 2001.

[69] H. Hyyrö. «Bit-Parallel LCS-length Computation Revisited». *In Proc. 15th Australasian Workshop on Combinatorial Algorithms (AWOCA)*. 2004, págs. 16-27.

[70] A. Benítez-Hidalgo, A. J. Nebro, J. García-Nieto, I. Oregi y J. Del Ser. «jMetalPy: A Python framework for multi-objective optimization with metaheuristics». *Swarm and Evolutionary Computation* 51 (2019), pág. 100598. ISSN: 2210-6502. DOI: https://doi.org/10.1016/j.swevo.2019.100598. URL: https://www.sciencedirect.com/science/article/pii/S2210650219301397.

[71] N. Kshetri. «Big data´s impact on privacy, security and consumer welfare». *Telecommunications Policy* 38.11 (2014), págs. 1134-1145.

[72] B. Thuraisingham. «Big data security and privacy». *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*. 2015, págs. 279-280.

[73] M. Kantarcioglu y E. Ferrari. «Research Challenges at the Intersection of Big Data, Security and Privacy». *Frontiers in Big Data* 2 (feb. de 2019), pág. 1. ISSN: 2624-909X. DOI: 10.3389/fdata.2019.00001.

[74] M. Poveda-Villalón, P. Espinoza-Arias, D. Garijo y O. Corcho. «Coming to Terms with FAIR Ontologies». *Knowledge Engineering and Knowledge Management*. Ed. por C. M. Keet y M. Dumontier. Cham: Springer International Publishing, 2020, págs. 255-270. ISBN: 978-3-030-61244-3.

[75] D. Garijo, O. Corcho y M. Poveda-Villalón. «FOOPS!: An Ontology Pitfall Scanner for the FAIR Principles». CEUR Workshop Proceedings 2980 (2021). URL: http://ceur-ws.org/Vol-2980/paper321.pdf.

[76] A. van den Berghe, R. Scandariato, K. Yskout y W. Joosen. «Design notations for secure software: a systematic literature review». *Software and Systems Modeling* 16 (3 jul. de 2017), págs. 809-831. ISSN: 16191374. DOI: 10.1007/s10270-015-0486-9.

[77] O. M.Surakhi, A. Hudaib, M. AlShraideh y M. Khanafseh. «A Survey on Design Methods for Secure Software Development». *International Journal of Computers & Technology* 16 (7 2017), págs. 7047-7064. DOI: 10.24297/ijct.v16i7.6467.

[78] J Viega. «Security in the Software Development Lifecycle: An introduction to {CLASP}, the Comprehensive Lightweight Application Security Process». *Secure Software, Inc., McLean, Virginia, USA, White Paper* (2005).

[79] M. Howard y S. Lipner. *The security development lifecycle: sdl: a process for developing demonstrably more secure software*. Microsoft Press, 2006, pág. 352. ISBN: 0735622140.

[80] G. McGraw. *Software Security: Building Security in*. Addison-Wesley Professional, 2006, pág. 6. ISBN: 0769526845. DOI: 10.1109/ISSRE.2006.43.

[81] O. Masmali y O. Badreddin. «Model Driven Security : A Systematic Mapping Study». *Software Engineering* 7 (2 2019), págs. 30-38.

[82] A. Mashkoor, A. Egyed y R. Wille. «Model-driven Engineering of Safety and Security Systems: A Systematic Mapping Study». *arXiv preprint arXiv:2004.08471* (2020). URL: http://arxiv.org/abs/2004.08471.

[83] H. Olivera, M. Holanda y F. Guimaraes. «Data modeling for NoSQL document-oriented databases». *Annual International Symposium on Information Management and Big Data (SIMBig)*. Vol. 1478 CEUR Workshop Proceedings. 2015, págs. 129-135.

[84] M. Chevalier, M. El Malki, A. Kopliku, O. Teste y R. Tournier. «Implementation of Multidimensional Databases with Document-Oriented NoSQL». *Big Data Analytics and Knowledge Discovery: 17th International Conference, DaWaK 2015, Valencia, Spain, September 1-4, 2015, Proceedings*. Ed. por S. Madria y T. Hara. Springer International Publishing, 2015, págs. 379-390. ISBN: 978-3-319-22729-0. DOI: 10.1007/978-3-319-22729-0_29.

[85] M. Chevalier, M. E. Malki, A. Kopliku, O. Teste y R. Tournier. «Implementation of Multidimensional Databases in Column-Oriented NoSQL Systems». *Advances in Databases and Information Systems: 19th East European Conference, ADBIS 2015, Poitiers, France, September 8-11, 2015, Proceedings*. Ed. por M. Tadeusz, P. Valduriez y L. Bellatreche. Springer International Publishing, 2015, págs. 79-91. ISBN: 978-3-319-23135-8. DOI: 10.1007/978-3-319-23135-8_6.

[86] Y. Li, P. Gu y C. Zhang. «Transforming UML class diagrams into HBase based on metamodel». Vol. 2. 2014, págs. 720-724. ISBN: 9781479931965. DOI: 10.1109/InfoSEEE.2014.6947760.

[87] D. Ruiz, S. Morales y J. Molina. «Inferring versioned schemas from NoSQL databases and its applications». *International Conference on Conceptual Modeling (ER)*. 2015, págs. 467-480.

[88] F. Bugiotti, L. Cabibbo, P. Atzeni y R. Torlone. «Database design for NoSQL systems». Vol. 8824. 2014, págs. 223-231. ISBN: 9783319122052. DOI: 10.1007/978-3-319-12206-9_18.

[89] S. Banerjee y A. Sarkar. «Modeling NoSQL databases: from conceptual to logical level design». 2016, págs. 10-12.

[90] D. B. Rawat, R. Doku y M. Garuba. «Cybersecurity in Big Data Era: From Securing Big Data to Data-Driven Security». *IEEE Transactions on Services Computing* (mar. de 2019), pág. 1. ISSN: 1939-1374. DOI: 10.1109/tsc.2019.2907247.

[91] N. Gupta y R. Agrawal. «Chapter Four - NoSQL Security». Ed. por P. Raj y G. C. Deka. Vol. 109. Elsevier, 2018, págs. 101-132. DOI: https://doi.org/10.1016/bs.adcom.2018.01.003. URL: https://www.sciencedirect.com/science/article/pii/S0065245818300032.

[92] B. Saraladevi, N. Pazhaniraja, P. V. Paul, M. S. S. Basha y P. Dhavachelvan. «Big data and Hadoop-A study in security perspective». Vol. 50. 2015. ISBN: 2992963984. DOI: 10.1016/j.procs.2015.04.091.

[93] I. L. Solsol, H. F. Vargas y G. M. Díaz. «Security Mechanisms in NoSQL DBMS's: A Technical Review». Ed. por F. R. Narváez, D. F. Vallejo, P. A. Morillo y J. R. Proaño. Springer International Publishing, 2020, págs. 215-228. ISBN: 978-3-030-46785-2.

[94] D. Pasqualin, G. Souza, E. L. Buratti, E. C. de Almeida, M. D. D. Fabro y D. Weingaertner. «A Case Study of the Aggregation Query Model in Read-Mostly NoSQL Document Stores». ACM Press, 2016, págs. 224-229. ISBN: 9781450341189. DOI: 10.1145/2938503.2938546.

[95] G. Weintraub y E. Gudes. «Data integrity verification in column-oriented NoSQL databases». Vol. 10980 LNCS. Springer Verlag, 2018, págs. 165-181. ISBN: 9783319957289. DOI: 10.1007/978-3-319-95729-6_11.

[96] W. Zugaj. «Analysis of Standard Security Features for Selected NoSQL Systems». *American Journal of Information Science and Technology* 3 (2 2019), pág. 41. ISSN: 2640-057X. DOI: 10.11648/j.ajist.20190302.12.

[97] S. Telghamti y L. Derdouri. «Towards a Trust-based Model for Access Control for Graph-Oriented Databases». 2021, págs. 1-3. DOI: 10.1109/ICTAACS53298.2021.9715180.

[98] M. Valzelli., A. Maurino. y M. Palmonari. «A Fine-grained Access Control Model for Knowledge Graphs». SciTePress, 2020, págs. 595-601. ISBN: 978-989-758-446-6. DOI: 10.5220/0009833505950601.

[99] C. Morgado, G. B. Baioco, T. Basso y R. Moraes. «A Security Model for Access Control in Graph-Oriented Databases». 2018, págs. 135-142. DOI: 10.1109/QRS.2018.00027.

[100] C. Blanco, D. García-Saiz, D. G. Rosado, A. S.-O. Parra, J. Peral, A. Maté, J. Trujillo y E. Fernández-Medina. «Security policies by design in NoSQL document databases.» *J. Inf. Secur. Appl.* 65 (2022), pág. 103120. DOI: 10.1016/j.jisa.2022.103120. URL: https://doi.org/10.1016/j.jisa.2022.103120.

[101] A. Maté, J. Peral, J. Trujillo, C. Blanco, D. García-Saiz y E. Fernández-Medina. «Improving security in NoSQL document databases through model-driven modernization». *Knowledge and Information Systems* (2021). ISSN: 0219-3116. DOI: 10.1007/s10115-021-01589-x.

[102] N. Konstantinou, D.-E. Spanos y N. Mitrou. «Ontology and Database Mapping: A Survey of Current Implementations and Future Directions.» *Journal of Web Engineering (JWE)* 7 (mar. de 2008), págs. 1-24.

[103] P. Mayadewi, B. Sitohang y F. Azizah. «Scheme mapping for relational database transformation to ontology: A survey». Nov. de 2017, págs. 1-6. DOI: 10.1109/ICODSE.2017.8285866.

[104] C. Ma y B. Molnár. «Ontology Learning from Relational Database: Opportunities for Semantic Information Integration». *Vietnam Journal of Computer Science* 9 (feb. de 2022), págs. 31-57. DOI: 10.1142/S219688882150024X.

[105] H. Abbes y F. Gargouri. «M2Onto: An Approach and a Tool to Learn OWL Ontology from MongoDB Database». *Intelligent Systems Design and Applications*. Ed. por A. M. Madureira, A. Abraham, D. Gamboa y P. Novais. Cham: Springer International Publishing, 2017, págs. 612-621. ISBN: 978-3-319-53480-0.

[106] H. Abbes, S. Boukettaya y F. Gargouri. «Learning ontology from Big Data through MongoDB database». *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. 2015, págs. 1-7. DOI: 10.1109/AICCSA.2015.7507166.

[107] V. K. Kiran y R Vijayakumar. «Ontology based data integration of NoSQL datastores». *2014 9th International Conference on Industrial and Information Systems (ICIIS)*. 2014, págs. 1-6. DOI: 10.1109/ICIINFS.2014.7036545.

[108] S. Ferilli. «Integration Strategy and Tool between Formal Ontology and Graph Database Technology». *Electronics* 10.21 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10212616. URL: https://www.mdpi.com/2079-9292/10/21/2616.

[109] N. Fathy, W. Gad, N. Badr y M. Hashem. «ProGOMap: Automatic Generation of Mappings From Property Graphs to Ontologies». *IEEE Access* 9 (2021), págs. 113100-113116. DOI: 10.1109/ACCESS.2021.3104293.

[110] C. Brewster, B. Nouwt, S. Raaijmakers y J. Verhoosel. «Ontology-based Access Control for FAIR Data». *Data Intelligence* 2.1-2 (ene. de 2020), págs. 66-77. ISSN: 2641-435X. DOI: 10.1162/dint_a_00029. eprint: https://direct.mit.edu/dint/article-pdf/2/1-2/66/1893368/dint_a_00029.pdf. URL: https://doi.org/10.1162/dint_a_00029.

[111] N. AbdulKadhim y M. Al-Wahah. «Semantic-Based Multi-Domain Data Access Authorization». *Journal of Physics: Conference Series* 1818.1 (mar. de 2021), pág. 012211. DOI: 10.1088/1742-6596/1818/1/012211. URL: https://doi.org/10.1088/1742-6596/1818/1/012211.

[112] F. Rosa y M. Jino. «A Survey of Security Assessment Ontologies». Mar. de 2017, págs. 166-173. ISBN: 978-3-319-56535-4. DOI: 10.1007/978-3-319-56535-4_17.

[113] R. Ferrini y E. Bertino. «Supporting RBAC with XACML+OWL». *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies*. SACMAT '09. New York, NY, USA: Association for Computing Machinery, 2009, 145–154. ISBN: 9781605585376. DOI: 10.1145/1542207.1542231. URL: https://doi.org/10.1145/1542207.1542231.

[114] T. Moses. «EXtensible Access Control Markup Language (XACML) version 1». *OASIS Standard* (ene. de 2005).

[115] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough y B. Thuraisingham. «ROWLBAC: representing role based access control in OWL». *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT* (jun. de 2008), págs. 73-82. DOI: 10.1145/1377836.1377849.

[116] N. Sharma y A. Joshi. «Representing Attribute Based Access Control Policies in OWL». Feb. de 2016, págs. 333-336. DOI: 10.1109/ICSC.2016.16.

[117] N. F. Noy y D. L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Inf. téc. Mar. de 2001. URL: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

[118] O. M. Group et al. *OMG XML Metadata Interchange (XMI) Specification. Version 2.0*. 2003. URL: https://www.omg.org/spec/XMI/2.0/.