Universidad de Valladolid

Escuela de Ingeniería Informática

de Valladolid

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

MENCIÓN EN COMPUTACIÓN

DataSecOnt: Ontología para la Seguridad de los Datos de Google Play

Alumno:

Gavino-Dias González, Alejandra

Tutora:

Martínez González, Mercedes

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que me han apoyado durante la realización de este trabajo.

En primer lugar, a mi tutora Mercedes por su confianza y por brindarme la oportunidad de formar parte de este proyecto. A Alejandro Pérez, por su ayuda continua.

A mi familia, por su apoyo incondicional a lo largo de mis estudios; a mi pareja, por estar siempre a mi lado y animarme siempre; y a mis compañeros de trabajo, que me han ayudado y acompañado incluso hablando otro idioma.

Resumen

Este trabajo tiene como objetivo el diseño, desarrollo e implementación de DataSecOnt, una ontología orientada a modelar de forma detallada el ciclo de vida de los datos en aplicaciones móviles, prestando especial atención a las finalidades de uso, la naturaleza de la información y las medidas de seguridad declaradas por los desarrolladores. La ontología se ha construido utilizando OWL en formato RDF/XML, conforme a los estándares establecidos por el W3C, y tomando como referencia la sección de Seguridad de los Datos de Google Play, que proporciona una clasificación oficial de categorías y prácticas de tratamiento de datos.

Para su validación, se ha implementado un repositorio funcional en GraphDB que permite ejecutar consultas SPARQL. Además, se han generado instancias reales sobre aplicaciones publicadas en Google Play, con el fin de ilustrar y comprobar la capacidad del modelo.

La verificación se ha llevado a cabo mediante un conjunto de Competency Questions y consultas complementarias, que han permitido evaluar la consistencia de la ontología y su capacidad de respuesta frente a los objetivos planteados. Entre los principales resultados destaca la correcta diferenciación entre datos recopilados y compartidos, la clasificación jerárquica de tipos de datos y la identificación de patrones de uso y medidas de seguridad, aspectos clave para fomentar la transparencia informativa y el cumplimiento normativo en el ámbito digital.

Finalmente, se proponen diversas líneas de trabajo futuro, entre las que se incluyen la integración con otras ontologías relacionadas, la ampliación del conjunto de instancias y el desarrollo de un agente inteligente capaz de interpretar consultas en lenguaje natural.

Palabras clave: ontologías, Web Semántica, OWL, RDF/XML, GraphDB, protección de datos, transparencia informativa, aplicaciones móviles.

Abstract

This work aims to design, develop, and implement DataSecOnt, an ontology intended to comprehensively model the data lifecycle in mobile applications, with a particular focus on purposes of use, the nature of the information, and the security measures declared by developers. The ontology has been built using OWL in RDF/XML format, following W3C standards and based on the Data Safety section of Google Play, which provides an official classification of data categories and processing practices.

For validation purposes, a functional repository was deployed in GraphDB, enabling the execution of SPARQL queries. Additionally, real-world instances were added from applications published on Google Play, in order to illustrate and test the model's capabilities.

Verification was conducted using a set of Competency Questions and complementary queries, which allowed the evaluation of the ontology's consistency and its ability to respond to the objectives defined. The main results highlight the correct differentiation between collected and shared data, the hierarchical classification of data types, and the identification of usage patterns and security measures—key aspects for promoting information transparency and regulatory compliance in the digital environment.

Finally, several lines of future work are proposed, including integration with related ontologies, the expansion of the dataset of instances, and the development of an intelligent agent capable of interpreting natural language queries

Keywords: ontologies, Semantic Web, OWL, RDF/XML, GraphDB, data protection, information transparency, mobile applications.

Índice

Índice de tablas

Índice de figuras

Capítulo 1 – Introducción

El presente trabajo de fin de grado corresponde a la memoria del desarrollo y explotación de una ontología OWL denominada DataSecOnt, diseñada para formalizar el conocimiento sobre la seguridad y privacidad de los datos en Google Play. A lo largo de los capítulos de esta memoria se explicarán la metodología, fases de implementación y resultados de la construcción y posterior análisis de la ontología DataSecOnt.

Este trabajo se incluye en las actividades del Proyecto Estratégico de Ciberseguridad "App-PI (App Privacy Impact): Un ecosistema para la evaluación del impacto de apps para dispositivos móviles sobre la privacidad y seguridad de sus usuarios", el cual se realiza al amparo de un convenio de colaboración entre la Universidad de Valladolid y la S.M.E. Instituto Nacional de Ciberseguridad de España M.P., S.A. para la promoción de proyectos estratégicos de Ciberseguridad en España, en el marco de los Fondos del Plan de Recuperación, Transformación y Resiliencia, financiados por la Unión Europea (Next Generation).

Referencia del proyecto: C120.23_Uva.

1.1 Contexto

1.1.1 Contexto general

La sección de Seguridad de Datos, se ha introducido como una medida para la protección de la privacidad del usuario prohibiendo las apps engañosas o que abusen de cualquier dato personal. La información proporcionada en la sección de Seguridad de Datos se muestra en el momento previo a la instalación de una aplicación, de esta forma, los usuarios pueden tomar una decisión informada sobre su privacidad. Esta iniciativa refleja una tendencia creciente en la industria hacia una mayor responsabilidad en el manejo de datos personales [1].

La complejidad de los datos que recopilan las aplicaciones, así como la diversidad de propósitos con los que estos pueden ser utilizados, han puesto de manifiesto la necesidad de herramientas más precisas para describir, categorizar y controlar esta información. No se trata únicamente de almacenar datos, sino de comprender qué representan, cómo se relacionan y bajo qué condiciones se pueden procesar. En este contexto, resulta fundamental contar con modelos que permitan estructurar de manera formal y coherente los distintos tipos de datos y sus características asociadas.

Asimismo, la existencia de normativas de protección de datos y los requisitos de transparencia ante los usuarios exigen un lenguaje común que facilite tanto a los desarrolladores como a las plataformas describir claramente qué información se gestiona y con qué fines. La construcción de representaciones compartidas y estandarizadas contribuye a reducir la ambigüedad y a mejorar la confianza en los sistemas [2].

Ontologías

Las ontologías tienen su origen en la filosofía, donde constituyen una rama que estudia la naturaleza del ser. Su objetivo es determinar qué entidades existen, cuáles son sus características esenciales y cómo se relacionan entre sí. Este enfoque filosófico ha servido de base para su aplicación en campos más técnicos y prácticos [3].

En el contexto de las ciencias de la información y la ingeniería del conocimiento, una ontología se define como una especificación formal de un conjunto de conceptos y de las relaciones que existen entre

ellos. Dicho de manera sencilla, es una forma estructurada de describir un área de conocimiento: define qué entidades existen en un dominio, qué propiedades tienen y cómo se interrelacionan [4].

Esta representación facilita que distintos sistemas puedan compartir información de manera coherente y sin ambigüedades.

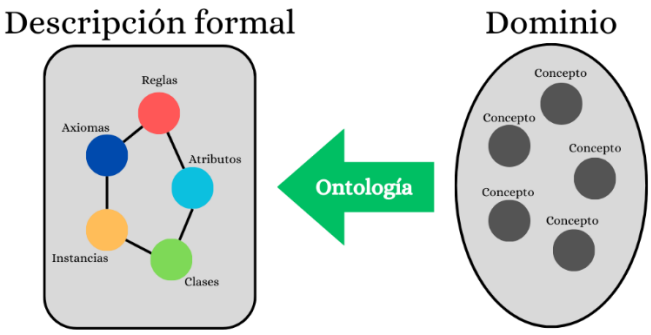

ILUSTRACIÓN 1: ESQUEMA DE UNA ONTOLOGÍA

Para construir una ontología, es necesario definir formalmente sus principales componentes:

- **Clases o conceptos**: representan categorías de objetos o ideas (por ejemplo, Usuario, Dispositivo, Dato Personal).
- **Individuos o instancias**: ejemplos concretos de esas clases (un usuario específico o un dato concreto).
- **Atributos o propiedades**: describen características de las clases (por ejemplo, nombre, tipo de dato).
- **Relaciones**: vinculan los conceptos entre sí (por ejemplo, un Usuario posee un Dispositivo).
- **Restricciones, axiomas y reglas**: establecen condiciones sobre cómo se pueden combinar o relacionar los elementos.

El proceso de modelado ontológico consiste en diseñar y construir estas representaciones formales. Para ello se utilizan lenguajes de ontologías, que proporcionan la sintaxis y la semántica necesarias para expresar conceptos y relaciones de forma precisa y procesable por máquinas.

Uno de los lenguajes más empleados es OWL (*Web Ontology Language*). Este lenguaje fue desarrollado por el consorcio W3C en el marco de la Web Semántica y está basado en la lógica descriptiva. OWL ofrece un conjunto de constructores que permiten [4, 5]:

- Definir jerarquías de clases (subclases y superclases).
- Especificar propiedades de datos y de objetos.
- Establecer restricciones sobre las relaciones (por ejemplo, cardinalidades).
- Formular axiomas que permitan realizar inferencias automáticas.

Ontologías en la gestión de información

En el ámbito de la gestión de información, las ontologías cumplen un papel fundamental, al proporcionar un marco común que facilita la organización, el intercambio y la reutilización del conocimiento. Estas representaciones formales describen de forma detallada los conceptos de un dominio, sus propiedades y las relaciones entre ellos, lo que las convierte en un recurso esencial para garantizar la interoperabilidad entre sistemas heterogéneos y la integración de datos provenientes de múltiples fuentes.

Las principales ventajas del uso de las ontologías son:

- **Facilitan la comunicación entre sistemas**: Al proporcionar un vocabulario común y estandarizar términos, las ontologías permiten que diferentes sistemas, aplicaciones o agentes inteligentes compartan y entiendan la información de manera consistente. Esto es especialmente útil en entornos donde coexisten múltiples fuentes de datos con diferentes formatos y estructuras.
- **Permiten razonamientos automáticos y consultas**: Gracias a la aplicación de reglas y relaciones definidas en la ontología, es posible realizar inferencias automáticas y consultas complejas.
- **Pueden actualizarse conforme evoluciona el conocimiento del dominio**: Las ontologías son flexibles y adaptables, lo que permite modificarlas y ampliarlas a medida que el conocimiento del dominio cambia o se expande, asegurando que la representación del conocimiento se mantenga actualizada y relevante.
- **Mejoran la integración de datos**: En entornos donde se manejan grandes volúmenes de información proveniente de diversas fuentes, las ontologías actúan como un puente que unifica y armoniza los datos.
- **Fomentan la reutilización del conocimiento**: Al proporcionar una representación formal y compartible, las ontologías permiten que el conocimiento generado en un contexto pueda ser reutilizado en otros proyectos o aplicaciones.
- **Soportan la toma de decisiones**: Al estructurar el conocimiento de manera lógica y accesible, las ontologías permiten a los sistemas y usuarios realizar análisis más profundos y tomar decisiones basadas en información precisa y bien organizada.

Las ontologías tienen aplicaciones muy diversas, desde la búsqueda semántica y la gestión del conocimiento, hasta su uso en sistemas de inteligencia artificial. Por ejemplo, son la base de sistemas expertos, asistentes virtuales o motores de recomendación, que utilizan el conocimiento formalizado para razonar e interactuar de forma más inteligente con los usuarios [6].

Gracias a su capacidad para organizar y representar información de forma precisa y adaptable, se han consolidado como una herramienta clave en un mundo en el que los datos son cada vez más abundantes y complejos. Su potencial para mejorar la comunicación, la integración y el análisis de información ponen en evidencia su importancia en ámbitos tan variados como la ciencia, la industria y la administración pública.

1.1.2 Contexto específico

La sección de Seguridad de los Datos en Google Play es una iniciativa que responde a la creciente preocupación de los usuarios por la transparencia en el tratamiento de su información personal y la necesidad de reforzar la confianza en las aplicaciones móviles. A través de este mecanismo, Google exige que los desarrolladores proporcionen información clara y detallada sobre diversos aspectos clave relacionados con la gestión de datos en sus aplicaciones. [1]

En concreto, los desarrolladores deben declarar:

- **La recopilación de datos y su propósito:** describiendo con precisión qué tipos de datos personales o sensibles recoge la aplicación (por ejemplo, ubicación, contactos, información financiera) y con qué finalidad se utilizan (como por ejemplo: mejora de la experiencia, personalización o análisis estadístico) .
- **El intercambio de datos con terceros:** informando si la aplicación comparte datos con otras empresas, proveedores de servicios o entidades externas, y en qué circunstancias.

- **Las prácticas de seguridad implementadas:** como las técnicas de cifrado de datos en tránsito y en reposo, las medidas de protección frente a accesos no autorizados y la existencia de procedimientos para que los usuarios puedan solicitar la eliminación de sus datos personales.
- **El cumplimiento de la Política de Familias de Google Play:** en caso de que la aplicación esté dirigida a menores o pueda ser utilizada por niños, se deben especificar las prácticas adicionales de protección de la privacidad infantil.
- **La validación frente al estándar MASVS (*Mobile Application Security Verification Standard*):** que permite a los desarrolladores demostrar que sus prácticas de seguridad han sido verificadas conforme a criterios reconocidos en la industria.

Desde el 20 de julio de 2022, esta sección es de carácter obligatorio para todas las aplicaciones que se publiquen o actualicen en Google Play. Además, la información proporcionada no queda únicamente en la declaración del desarrollador: Google se reserva el derecho de revisar y validar la exactitud de los datos, y en caso de incumplimiento, puede adoptar medidas que incluyen la suspensión o eliminación de la aplicación de la plataforma [1,7].

Esta medida facilita que los usuarios dispongan de información clara y actualizada sobre cómo se gestionan sus datos, permitiéndoles tomar decisiones más informadas antes de instalar una aplicación y fortaleciendo la seguridad del ecosistema Android. Estas iniciativas en privacidad y protección de datos reflejan una tendencia global que también ha impulsado la necesidad de describir de manera precisa los datos recopilados y sus finalidades, de modo que la información sea comprensible y verificable. En este contexto, regulaciones internacionales como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea han establecido principios y obligaciones que han contribuido a elevar los estándares de transparencia y responsabilidad en el uso de los datos personales. Este reglamento, de aplicación obligatoria desde 2018, define aspectos fundamentales como la necesidad de informar de forma clara y accesible sobre el tratamiento de los datos, la obtención del consentimiento explícito del usuario, el derecho a la supresión y la portabilidad de la información, así como la responsabilidad proactiva de los responsables del tratamiento. Estas exigencias normativas han influido directamente en el desarrollo de iniciativas como la sección de Seguridad de los Datos en Google Play. En este sentido, la construcción de modelos conceptuales estructurados, como las ontologías, se convierte en una herramienta esencial para garantizar la transparencia y facilitar el cumplimiento de estos requisitos [2,7].

1.1.3 Presentación del problema

A pesar de que la sección Seguridad de los Datos de Google Play establece una estructura predefinida para describir la recogida y el uso de la información personal, la diversidad de aplicaciones, categorías y prácticas declaradas genera un volumen de datos complejo y heterogéneo. Esta variedad dificulta su análisis sistemático y la comparación entre aplicaciones o grupos de datos.

Para afrontar este problema, es importante contar con un modelo que permita representar de forma clara y estructurada la información recogida, de manera que sea más fácil interpretarla, procesarla y reutilizarla. Es por ello que desarrollo de una ontología específica se plantea como una solución adecuada para organizar este conocimiento y ofrecer un marco conceptual bien definido.

1.2 Motivación

Las tendencias crecientes en seguridad móvil están estrechamente relacionadas con la necesidad de una ontología como DataSecOnt. El auge del uso de la inteligencia artificial y del internet de las cosas en aplicaciones móviles ha convertido la mejora de la privacidad de los datos en una prioridad fundamental.

Podemos encontrar casos recientes que ilustran la magnitud de este problema:

- **Facebook (Meta)**: En 2021, datos personales de más de 533 millones de usuarios (números de teléfono, nombres completos, ubicaciones y direcciones de correo electrónico) fueron filtrados debido a una vulnerabilidad en su sistema, estos datos seguían siendo explotados en 2023 [8].
- **TikTok**: En 2022, investigadores descubrieron que esta aplicación almacenaba datos sensibles de usuarios en servidores ubicados en China, lo que generó preocupaciones sobre la transferencia internacional de datos y el acceso potencial por parte del gobierno chino. Planteando dudas sobre la transparencia en el manejo de los datos [9].
- **Google Play**: En 2023, Google bloqueó la publicación de más de 2,28 millones de aplicaciones maliciosas que recopilaban datos personales sin consentimiento, como números IMEI, ubicaciones GPS y contactos. Estas aplicaciones, que se presentaban como herramientas legítimas, incluían código que enviaba información a servidores externos sin conocimiento del usuario [10].

Todo este contexto evidencia que, más allá de las medidas de seguridad técnicas, es esencial garantizar que la información sobre qué datos se recogen, con qué finalidad y cómo se protegen sea clara, accesible y verificable. Además, regulaciones internacionales como el Reglamento General de Protección de Datos (GDPR) obligan a las organizaciones a informar de forma transparente sobre el tratamiento de los datos personales y a cumplir principios como la minimización, la exactitud y la limitación de la finalidad. Estas exigencias, unidas al creciente volumen de datos y a la complejidad de los entornos móviles, refuerzan la importancia de contar con mecanismos que permitan mejorar la transparencia, la trazabilidad y la confianza de los usuarios [2].

Además, una ontología de estas características puede tener aplicaciones prácticas muy diversas. Entre ellas, cabe destacar su utilidad como soporte en procesos de auditoría y supervisión de aplicaciones, al permitir verificar de forma sistemática qué datos se declaran y cómo se gestionan. También puede servir como recurso de referencia para grupos de investigación y actividades de formación, facilitando tanto el estudio en profundidad de la privacidad y la seguridad en entornos móviles como la explicación de estos conceptos de manera estructurada en contextos educativos o de divulgación. Asimismo, puede constituir la base para el desarrollo de herramientas que automaticen la comprobación del cumplimiento normativo o permitan comparar aplicaciones en función de sus prácticas de gestión de datos.

1.3 Objetivos

DataSecOnt es una propuesta de ontología desarrollada en OWL que tiene como propósito principal facilitar la representación estructurada y el análisis de la información relativa a la seguridad de los datos en aplicaciones móviles. En particular, se centra en formalizar el conocimiento sobre las categorías, tipologías y prácticas en relación con los datos declaradas por los desarrolladores en Google Play, con el fin de contribuir a una mayor transparencia y comprensión de estos aspectos.

Este proyecto se plantea tanto como una iniciativa de aplicación práctica de tecnologías semánticas como una aportación académica orientada a demostrar el potencial de las ontologías en el ámbito de la privacidad y la protección de datos.

A continuación, se detallan los objetivos académicos y de desarrollo que guían este Trabajo de Fin de Grado.

1.3.1 Objetivos académicos

Este Trabajo de Fin de Grado persigue objetivos de carácter académico relacionados con la aplicación práctica de tecnologías semánticas y con la documentación rigurosa del proceso. A continuación, se describen los principales objetivos académicos planteados.

- **Facilitar la inferencia de nuevo conocimiento a partir de datos estructurados:**
  Uno de los objetivos principales es demostrar cómo una ontología bien diseñada permite generar conocimiento implícito a partir de información explícita. A través de axiomas, jerarquías y relaciones semánticas, se busca que el modelo sea capaz de inferir automáticamente hechos adicionales mediante el uso de razonadores OWL.

- **Adquirir un conocimiento práctico y profundo de las tecnologías semánticas y las herramientas de modelado:**
  El proyecto también persigue familiarizarse con el uso de lenguajes y estándares de la Web Semántica (RDF, OWL y SPARQL), así como con plataformas especializadas como *GraphDB*. El objetivo es alcanzar un nivel de competencia que permita manejar estas tecnologías con solvencia en proyectos reales y comprender sus posibilidades y limitaciones.

- **Evaluar la capacidad explicativa de la ontología a través de preguntas de competencia:**
  A través del planteamiento y resolución de *Competency Questions* (preguntas de competencia), se busca comprobar que la ontología desarrollada cubre de manera efectiva el conocimiento del dominio. Estas preguntas actúan como criterio de validación formal del modelo. El objetivo académico es, por tanto, doble: por un lado, construir una ontología coherente y por otro, comprobar su capacidad para responder a consultas mediante SPARQL, utilizando datos reales.

- **Garantizar la trazabilidad y la justificación metodológica del desarrollo de la ontología:**
  Este trabajo tiene también como objetivo fundamental documentar de forma clara todas las fases del desarrollo de la ontología, desde el análisis de requisitos hasta su implementación técnica. La documentación seguirá criterios académicos y metodológicos rigurosos, de manera que se pueda garantizar la trazabilidad del proceso, la reproducibilidad del modelo y la reutilización del conocimiento generado, tanto con fines docentes como de investigación.

1.3.2 Objetivos de desarrollo

Desde una perspectiva técnica, este proyecto tiene como objetivo principal la construcción de una ontología funcional y semánticamente coherente, capaz de representar y analizar la información proporcionada en la sección de Seguridad de los Datos de Google Play. Para ello, se establecen los siguientes objetivos específicos de desarrollo:

- **Formalizar el conocimiento sobre las categorías y tipologías de datos tratadas en Google Play:**
  Se busca transformar el contenido informal y heterogéneo de las descripciones que los desarrolladores proporcionan sobre el tratamiento de datos en Google Play en un modelo semántico estructurado y formal. Esta formalización incluye la identificación de clases, propiedades, relaciones jerárquicas y restricciones que permitan representar de manera precisa los distintos tipos de datos (personales, financieros, de ubicación, etc.), sus categorías, finalidades de uso, mecanismos de manejo y medidas de seguridad asociadas.

- **Implementar la ontología utilizando herramientas estándares como GraphDB y OWL:**

El desarrollo técnico de la ontología se llevará a cabo con herramientas especializadas en modelado semántico. Se empleará OWL (*Web Ontology Language*) como lenguaje formal para definir la estructura conceptual del dominio, mientras que GraphDB se utilizará como plataforma para el almacenamiento, razonamiento y consulta del modelo.

- **Definir un conjunto de preguntas de competencia como criterio de validación del modelo ontológico:**

  El objetivo es formular un conjunto representativo y bien estructurado de estas preguntas, alineadas con los escenarios de uso esperados, y usarlas como guía para orientar las decisiones de modelado durante el diseño conceptual e implementación. Estas preguntas también constituirán una base fundamental para la validación funcional del sistema.

- **Diseñar consultas semánticas que faciliten la explotación y verificación de la ontología:**

  Se pretende crear un conjunto de consultas SPARQL que cubran tanto las *Competency Questions* como otros posibles escenarios. Estas consultas permitirán validar que la ontología responde correctamente a las necesidades del dominio y que es capaz de generar información relevante para el análisis de aplicaciones móviles desde el punto de vista de la privacidad y la seguridad de los datos.

1.4 Estructura de la memoria

El presente documento se estructura en diez capítulos, ordenados de forma lógica y progresiva, que abarcan desde la introducción teórica del problema hasta el desarrollo práctico de la ontología y la evaluación de sus resultados. A continuación, se describen brevemente los contenidos de cada uno de ellos:

- **Capítulo 1 – Introducción**: Presenta el contexto general y específico del problema, así como la motivación que ha llevado al desarrollo de este trabajo. También se explicitan los objetivos académicos y de desarrollo que guían el proyecto, y se ofrece una visión global de la organización del documento.
- **Capítulo 2 – Metodología**: Describe el enfoque metodológico adoptado para la construcción de la ontología, basado en la guía *Ontology Development 101* y *Ontology Requirement Specification Document* (ORSD). Se detallan los pasos seguidos y la justificación para esta combinación de metodologías.
- **Capítulo 3 – Requisitos y análisis**: Reúne la información necesaria para delimitar el dominio y formalizar los requisitos del sistema. Se especifica el objetivo principal de la ontología, se formulan las *Competency Questions*, y se establecen los requisitos funcionales y no funcionales. También se identifican las fuentes de información utilizadas, se definen los conceptos clave y se analizan los riesgos asociados al desarrollo.
- **Capítulo 4 – Diseño conceptual**: Expone el diseño ontológico. Se detallan las clases principales y su jerarquía, así como las relaciones semánticas y propiedades que definen el comportamiento del modelo. Este capítulo sienta las bases formales para la implementación posterior.
- **Capítulo 5 – Implementación**: Describe el proceso técnico de codificación de la ontología utilizando tecnologías semánticas como OWL, RDF y SPARQL. Se incluyen el diseño de URIs, la estructuración del modelo en RDF/XML, la declaración formal de la ontología, y la creación de ejemplos de instancias.
- **Capítulo 6 – Validación y evaluación**: Presenta las técnicas de verificación utilizadas para asegurar la consistencia lógica de la ontología. Se evalúa la capacidad del modelo para

responder a las *Competency Questions* mediante consultas SPARQL, y se identifican y documentan posibles mejoras o refinamientos.

- **Capítulo 7 – Resultados y análisis**: Resume los principales resultados obtenidos tras la implementación, validación y explotación de la ontología. Asimismo, se analizan las evidencias extraídas a través de las consultas.
- **Capítulo 8 – Conclusiones y líneas de trabajo futuras**: Expone las conclusiones generales del proyecto, diferenciando entre los aprendizajes académicos y los logros de desarrollo. Finalmente, se proponen posibles líneas de mejora o ampliación para futuros trabajos en este ámbito.
- **Capítulo 9 – Bibliografía**: Recoge todas las fuentes bibliográficas consultadas a lo largo del proyecto, siguiendo un formato estandarizado.
- **Capítulo 10 – Anexos**: Incluye materiales complementarios como representaciones completas del modelo RDF/XML, ejemplos de consultas SPARQL, tablas de clasificación y cualquier otro contenido de apoyo necesario para la comprensión o reproducción del trabajo. También se presenta el entorno de desarrollo utilizado (GraphDB) y se documenta paso a paso el proceso de implementación.

1.5 Estado del arte y herramientas

En el ámbito de la representación semántica de la privacidad y la gestión de datos personales, se han desarrollado diversas iniciativas que proponen modelos y ontologías específicas. Entre ellas destacan el Data Privacy Vocabulary (DPV) [11], promovido por el W3C, que define un vocabulario para describir finalidades, bases legales y tipos de datos personales, y la ontología PPO (Privacy Preference Ontology) [12], centrada en modelar preferencias de privacidad de los usuarios.

De forma destacada, se tomó como referencia la ontología desarrollada por el grupo App-PI, que modela los permisos solicitados por aplicaciones en Google Play, incluyendo aspectos como los tipos de permiso, las categorías de riesgo y el acceso a funciones del dispositivo. No obstante, mientras dicha ontología se centra principalmente en describir los permisos de las aplicaciones, este proyecto aborda de manera complementaria la representación de las categorías y tipologías de datos declaradas en la sección de Seguridad de los Datos, con el objetivo de cubrir un aspecto específico que no se encuentra desarrollado en detalle en trabajos previos [13].

1.5.1 Herramientas

En este apartado se describen las tecnologías y herramientas utilizadas para el desarrollo, representación e implementación del proyecto y la ontología DataSecOnt.

A continuación, se detallan los principales recursos utilizados: RDF como lenguaje base para la codificación de la ontología, OWL como marco semántico para definir restricciones y estructuras jerárquicas, GraphDB como motor de almacenamiento y razonamiento, SPARQL como lenguaje de consulta sobre el modelo, y Draw.io como herramienta de apoyo para la creación de esquemas y diagramas incluidos en distintas secciones de la memoria.

5.5.1 RDF

Para la representación formal de la ontología DataSecOnt se ha utilizado RDF (Resource Description Framework) como base tecnológica. RDF es una especificación del W3C diseñada para describir información estructurada mediante expresiones del tipo sujeto–predicado–objeto, también conocidas como tripletas RDF [14].

Este enfoque permite modelar entidades, relaciones y valores de manera flexible, reutilizable e interoperable. RDF es independiente del dominio y se adapta perfectamente a contextos semánticos como el tratamiento de datos personales en aplicaciones móviles.

Además, RDF permite representar conocimiento en distintos formatos de serialización. En este trabajo se ha utilizado el formato RDF/XML, ampliamente soportado por herramientas como *Protégé* y GraphDB.

5.5.2 OWL

Complementando a RDF, la ontología ha sido modelada utilizando OWL (*Web Ontology Language*), un lenguaje formal diseñado para definir la semántica de clases, propiedades y restricciones dentro de un dominio específico. OWL amplía las capacidades de RDF al proporcionar una mayor expresividad, permitiendo describir relaciones más complejas y especificar restricciones lógicas sobre los datos [4].

En concreto, OWL permite expresar aspectos como:

- Jerarquías de clases (rdfs:subClassOf).
- Tipos de propiedades (de objeto y de datos).
- Restricciones de cardinalidad (owl:cardinality, minCardinality), dominios y rangos.
- Relación entre clases (owl:equivalentClass, owl:disjointWith).
- Reglas que pueden ser interpretadas por razonadores para realizar inferencias automáticas.

OWL2 define diferentes perfiles o sub-lenguajes que limitan deliberadamente algunas capacidades de expresividad a cambio de mejorar el rendimiento computacional y la escalabilidad. En este proyecto se ha utilizado el perfil OWL2 RL (*Rule Language*), pensado para escenarios donde se requiere un razonamiento eficiente mediante reglas derivadas de la ontología. Este perfil permite representar la mayoría de las construcciones básicas de OWL manteniendo un nivel de complejidad reducido, lo que facilita su implementación en motores de inferencia basados en reglas y hace posible procesar grandes volúmenes de datos de manera ágil [15].

La elección de OWL2 RL responde a la necesidad de equilibrar expresividad y eficiencia computacional, ya que el modelo debía ser capaz de definir restricciones como la obligatoriedad de ciertas propiedades o la clasificación automática de los tipos de datos sensibles, pero también garantizar tiempos de respuesta razonables al ejecutar inferencias y consultas sobre el grafo. OWL, en combinación con RDF y SPARQL, constituye así la base tecnológica para representar el conocimiento de forma estructurada, coherente y verificable en la ontología DataSecOnt.

5.5.3 GraphDB

GraphDB es un sistema de gestión de bases de datos triplestore desarrollado por Ontotext, orientado al tratamiento de datos RDF y ontologías OWL [15].

Entre sus principales funcionalidades se encuentran:

- Almacenar triples RDF
- Aplicar razonamiento semántico sobre OWL
- Ejecutar consultas SPARQL
- Visualizar jerarquías y relaciones

Una de las características más destacadas de GraphDB es su capacidad para inferir conocimiento nuevo a partir de las axiomas OWL definidos en la ontología. Por ejemplo, si se establece que Correo es subclase de DatoMensaje y una aplicación recopila Correo, el sistema puede deducir automáticamente que también recopila un TipoDeDato, sin que esta relación esté explícitamente declarada. Esta funcionalidad resulta especialmente útil para validar la coherencia del modelo y enriquecer los resultados de las consultas.

El sistema se utiliza a través de una interfaz web denominada Workbench, que facilita la creación de repositorios, la carga de archivos en diversos formatos (RDF/XML, Turtle, JSON-LD, entre otros)

y la ejecución de consultas SPARQL sobre los grafos RDF cargados. También ofrece una vista estructurada de las clases y propiedades, facilitando la navegación y análisis del modelo.

En este trabajo se ha utilizado GraphDB Free Edition, versión gratuita suficiente para fines académicos. El razonador ha sido configurado con el perfil OWL2 RL, idóneo para realizar inferencias eficientes sobre jerarquías, dominios, rangos y restricciones, como las definidas en la ontología DataSecOnt [15].

5.5.3.1 Entorno de acceso remoto a GraphDB

Para poder utilizar GraphDB en condiciones óptimas y contar con un entorno gráfico compatible, se habilitó una máquina virtual con sistema operativo Ubuntu configurada específicamente para este propósito. Esta máquina, provista por la universidad, sirvió como entorno aislado y controlado para el despliegue de GraphDB y la gestión de la ontología RDF.

Dado que se accedía a la máquina virtual únicamente mediante la línea de comandos (SSH), y GraphDB requiere entorno gráfico para utilizar su interfaz web (Workbench), fue necesario habilitar el acceso remoto mediante Google Remote Desktop. Esta herramienta permitió conectar de forma segura y fluida al entorno de escritorio de la máquina virtual desde un navegador web, facilitando así el uso completo de GraphDB sin necesidad de instalar navegadores gráficos directamente en el sistema huésped [16].

Gracias a este sistema, fue posible realizar las tareas de validación semántica, ejecución de consultas SPARQL y navegación por el grafo RDF directamente desde Workbench de GraphDB, tal y como se haría en un entorno de escritorio local. Esta infraestructura remota resultó clave para garantizar la accesibilidad a GraphDB durante todo el desarrollo del proyecto.

5.5.4 SPARQL

SPARQL es el lenguaje estándar recomendado por el W3C para realizar consultas sobre datos estructurados en RDF. Su sintaxis está inspirada en SQL, pero adaptada al modelo de grafos RDF, permitiendo recuperar información mediante patrones de tripletas [14].

En el contexto de este trabajo, se ha utilizado SPARQL para realizar consultas sobre la ontología DataSecOnt cargada en GraphDB, y comprobar si el modelo es capaz de responder a las preguntas de competencia definidas en las fases previas.

Gracias al soporte de inferencia de GraphDB y las reglas OWL definidas, SPARQL permite no solo recuperar datos explícitos, sino también aquellos inferidos automáticamente. Esto lo convierte en una herramienta esencial para validar el modelo, extraer conocimiento y demostrar su aplicabilidad en escenarios reales [17].

Ejemplo de consulta SPARQL:

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT ?app ?dato
WHERE {
 ?app a ds:Aplicacion ;
     ds:recopilaDato ?dato .
 ?dato ds:datoEsSensitivo true .
}
```

Esta consulta SPARQL tiene como objetivo identificar qué aplicaciones recopilan datos considerados sensibles. Para ello, busca todas las instancias de ds:Aplicacion que estén relacionadas

mediante la propiedad ds:recopilaDato con algún TipoDeDato cuya propiedad ds:datoEsSensitivo tenga valor true.

La estructura de la consulta es la siguiente:

- ?app: variable que representa la aplicación. Se filtra por tipo (a ds:Aplicacion).
- ?dato: variable que representa el tipo de dato recopilado.

Se establece que ese dato debe estar relacionado con la aplicación (ds:recopilaDato ?dato) y debe tener la marca de sensitivo (ds:datoEsSensitivo true).

SPARQL ha resultado una herramienta clave para comprobar que la ontología funciona correctamente y es capaz de responder a las preguntas que motivaron su desarrollo. Gracias a su versatilidad y al soporte de inferencia de GraphDB, ha sido posible extraer información de manera clara y estructurada mediante consultas.

5.5.5 Draw.io

Durante el desarrollo del proyecto se ha utilizado draw.io (actualmente conocido también como diagrams.net) como herramienta principal para realizar diagramas. Esta aplicación web gratuita y multiplataforma ha permitido representar de manera clara y estructurada distintos elementos clave del trabajo, sirviendo como apoyo tanto en la fase de planificación como en la documentación final [18].

En particular, draw.io ha sido empleada para elaborar los siguientes tipos de diagramas:

- **Diagrama WBS:** Para descomponer el proyecto en tareas jerárquicas y facilitar una visión modular del trabajo a realizar. Esta representación ha sido clave en la planificación técnica, permitiendo organizar las tareas de forma estructurada y comprensible.
- **Diagrama de Gantt:** Ha sido utilizado para representar el cronograma del proyecto, identificando las fases, fechas previstas y dependencias entre actividades. Aunque se ha representado de forma estática, su creación en draw.io ha facilitado una visualización clara del calendario de desarrollo.
- **Diagramas conceptuales:** Se han diseñado esquemas para ilustrar la estructura lógica de la ontología, incluyendo las clases principales, relaciones y propiedades más relevantes. Estos diagramas conceptuales han sido de gran ayuda tanto para el diseño inicial como para la validación de la estructura del modelo ontológico.
- **Relaciones jerárquicas entre conceptos:** La herramienta también ha permitido representar visualmente jerarquías de clases y subclases, facilitando la comprensión de las relaciones taxonómicas y la organización general de la ontología.

La elección de draw.io se ha basado en su facilidad de uso, flexibilidad, y capacidad para exportar diagramas en distintos formatos (imagen, PDF, SVG, etc.), lo cual ha sido especialmente útil para su inclusión en esta memoria. Además, su funcionalidad colaborativa ha permitido iterar fácilmente sobre los diagramas, facilitando su revisión y mejora durante el desarrollo del proyecto.

Capítulo 2 – Metodología

No existe una única forma correcta de modelar un dominio, ya que la mejor solución depende de las particularidades del problema abordado. Sin embargo, toda metodología ontológica debe garantizar que los elementos del modelo estén estrechamente vinculados a los conceptos y relaciones del dominio, los cuales suelen expresarse a través de sustantivos y verbos presentes en las descripciones del mismo.

El desarrollo de la ontología DataSecOnt se ha basado en la integración de dos metodologías complementarias y ampliamente reconocidas en el ámbito del desarrollo de ontologías: por un lado, la especificación de requisitos mediante el enfoque ORSD [19] y, por otro, el desarrollo estructurado del modelo conceptual a través de *Ontology Development 101* [6].

Ambas metodologías se combinan para cubrir de forma sistemática las dos fases esenciales del desarrollo de una ontología: la definición de requisitos y el diseño/implementación del modelo. A continuación, se describen sus características y la motivación de su elección.

2.1 Ontology Development 101

La metodología utilizada en este proyecto se basa en la guía "*Ontology Development 101: A Guide to Creating Your First Ontology*", publicada por Natalya F. Noy y Deborah L. McGuinness en 2001 en el marco del Stanford Knowledge Systems Laboratory y el Stanford Medical Informatics [6]. Este enfoque se ha consolidado como una de las referencias más conocidas y utilizadas para la construcción de ontologías, gracias a su carácter práctico, adaptable y orientado a resultados.

Se trata de un proceso iterativo y sistemático que facilita la identificación, organización y formalización de los conceptos más relevantes de un dominio, proporcionando un conjunto de pasos claros que permiten avanzar desde la definición inicial del alcance hasta la creación de instancias.

En este proyecto, se han seguido estos pasos de forma progresiva, adaptándolos a las particularidades del dominio de seguridad de datos en aplicaciones móviles. A continuación, se describen las fases aplicadas:

2.1.1 Determinar el dominio y el alcance de la metodología

El primer paso consiste en delimitar con precisión el dominio de conocimiento que se desea representar. Para ello, se responde a preguntas clave como:

- ¿Qué dominio cubrirá la ontología?
- ¿Qué tipo de información debe representar y procesar?
- ¿Qué tipo de preguntas debe responder la ontología?
- ¿Quiénes serán los usuarios finales y responsables del mantenimiento?

Definir correctamente el dominio y el alcance resulta esencial para evitar que la ontología crezca de manera descontrolada o que incluya conceptos irrelevantes. Por ejemplo, una ontología orientada a describir componentes de un sistema informático podría limitarse a aspectos de hardware o incluir también elementos de software y redes, según el alcance establecido. De la misma forma, una ontología sobre datos personales puede centrarse únicamente en categorías de información, en políticas de tratamiento o en ambos aspectos combinados.

2.1.2 Considerar reutilizar una ontología existente

Antes de construir una ontología desde cero, es recomendable analizar ontologías ya existentes que puedan ser reutilizadas parcial o totalmente. Este enfoque promueve la interoperabilidad y reduce la duplicación de esfuerzos.

La reutilización puede adoptar diferentes formas, como:

- Incorporar directamente clases, propiedades o axiomas de una ontología existente.
- Extender una ontología de referencia añadiendo conceptos específicos del nuevo dominio.
- Integrar varios modelos parciales en una ontología combinada.

Este paso es especialmente relevante cuando se trabaja en áreas con estándares consolidados o con vocabularios ampliamente aceptados, ya que facilita la compatibilidad y el intercambio de información entre aplicaciones. No obstante, es importante analizar cuidadosamente si las ontologías existentes se ajustan al alcance definido, si su nivel de detalle es adecuado y si las licencias de uso permiten su incorporación.

En caso de que no exista un modelo que cumpla con los requisitos, se puede optar por diseñar una ontología nueva, adaptada específicamente a las necesidades del proyecto.

2.1.3 Enumeración de términos importantes en la ontología

Se elabora un listado exhaustivo de términos representativos del dominio, sin preocuparse inicialmente por su estructura jerárquica o relaciones internas. Esta lista funciona como base conceptual para la definición posterior de clases, propiedades y axiomas.

Podemos construir dicha lista respondiendo a las preguntas:

- ¿Qué términos nos gustaría usar?
- ¿Qué propiedades tienen esos términos?
- ¿Qué podemos decir de estos términos?

Se incluyen sustantivos clave, verbos representativos de acciones o relaciones, y atributos relevantes de los elementos descritos en la documentación oficial de Google Play.

Inicialmente, es importante obtener una lista completa de términos sin preocuparse por el solapamiento entre los conceptos que representan, las relaciones entre los términos, o las propiedades que puedan tener los conceptos.

2.1.4 Definición de las clases y la jerarquía entre clases

Tenemos varias aproximaciones para el desarrollo de este paso [20]:

1. ***Top-down* (de arriba abajo)**: comienza con la definición de los conceptos más generales del dominio y la posterior especialización del concepto.
2. ***Bottom-up* (de abajo a arriba)**: parte de la definición de las clases más específicas, las hojas de la jerarquía, con la posterior agrupación de estas clases en conceptos más generales.
3. **Combinación de ambas**: Definimos conceptos más importantes y especializamos y generalizamos para completar la ontología.

Ninguna de ellas es necesariamente mejor y habitualmente los conceptos de nivel intermedio son los más manejables, ya que presentan cierta estructura al contrario que los de bajo nivel.

2.1.5 Definición de las propiedades de las clases

Una vez establecidas las clases principales, es necesario describir su estructura interna mediante la definición de propiedades. Estas propiedades permiten especificar qué información caracteriza a cada clase y cómo se relaciona con otras entidades del modelo.

En esta fase se determina una lista de atributos y relaciones que serán fundamentales para responder de manera precisa a las preguntas de competencia identificadas previamente. Las propiedades se pueden clasificar en diferentes tipos, según su función:

- **Propiedades descriptivas**: expresan cualidades o características de la clase (por ejemplo, fecha de creación, estado, nivel de sensibilidad).
- **Propiedades identificadoras**: permiten distinguir de forma única cada instancia (por ejemplo, nombre, ID).
- **Propiedades de composición o partes**: representan componentes que forman parte de una entidad (por ejemplo, contiene, incluye).
- **Relaciones con otras clases**: establecen vínculos con instancias de clases distintas (por ejemplo, pertenece a, está asociado con).

Esta definición detallada de propiedades es esencial para garantizar que la ontología sea lo suficientemente rica y precisa, y que pueda soportar razonamiento semántico y validación de los datos representados.

2.1.6 Definición de las características de las propiedades

Una vez identificadas las propiedades, es necesario especificar con detalle sus características y restricciones. Este paso es fundamental para garantizar la coherencia del modelo y asegurar que las instancias creadas cumplan los requisitos establecidos.

Entre los aspectos que se definen habitualmente se encuentran:

- **Cardinalidad**: el número de valores permitidos para cada propiedad, indicando los límites mínimo y máximo de ocurrencias (por ejemplo, una aplicación puede tener como mínimo un identificador y como máximo un nombre oficial).
- **Tipo de valor aceptado**: determina la naturaleza de los datos que se pueden asignar a la propiedad (por ejemplo, texto, número, valor booleano).
- **Dominio y rango**: el dominio especifica a qué clases se aplica la propiedad, mientras que el rango indica qué tipo de valor o clase puede tomar. El dominio debe ser suficientemente amplio para cubrir los casos necesarios, pero sin perder especificidad.
- **Obligatoriedad**: se indica si la propiedad es obligatoria u opcional en las instancias de la clase.

Estas características se definen y modelan en OWL mediante restricciones lógicas y axiomas, que permiten validar automáticamente los datos y realizar inferencias sobre la información representada.

2.1.7 Creación de instancias

El último paso del proceso consiste en definir instancias individuales de las clases incluidas en la ontología. Estas instancias representan ejemplos concretos de los conceptos modelados y sirven para validar que la estructura y las restricciones establecidas funcionan correctamente.

Cada instancia se asocia a un conjunto de propiedades y relaciones que describen sus características y vinculan el dato con otros elementos del dominio. Este proceso de creación de ejemplos prácticos permite comprobar:

- Que las restricciones de cardinalidad, dominio y rango se cumplen.
- Que las propiedades obligatorias están correctamente definidas.

- Que la ontología es capaz de responder adecuadamente a las preguntas de competencia formuladas durante el diseño.

Además, las instancias facilitan la detección de inconsistencias o ambigüedades en la definición de clases y propiedades, contribuyendo a mejorar la calidad final del modelo. La creación de instancias también es un paso importante para demostrar la aplicabilidad de la ontología en situaciones reales.

2.2 *Ontology Requirements Specification Document* (ORSD)

Esta metodología proporciona una estructura clara y validada para definir el propósito, el alcance, los requisitos funcionales y no funcionales, así como las fuentes de información y los términos clave del dominio a modelar[19].

La elección de ORSD se fundamenta en su capacidad para:

- Sistematizar la identificación de necesidades y restricciones del dominio.
- Estructurar la recogida y validación de requisitos ontológicos.
- Favorecer la alineación con estándares internacionales como OWL y RDF.
- Facilitar la validación posterior del modelo mediante *Competency Questions*.

El documento ORSD se compone de los siguientes bloques principales:

2.2.1 Propósito

El objetivo de la metodología ORSD es definir con precisión los requerimientos del sistema ontológico antes de su implementación, asegurando que el modelo represente fielmente el conocimiento del dominio y pueda ser validado formalmente.

2.2.2 Alcance

El enfoque de ORSD delimita el contenido temático de la ontología, estableciendo claramente qué se va a representar (por ejemplo, prácticas de recopilación y compartición de datos en apps móviles) y qué usos se prevé que tenga la ontología. Esto permite enfocar el diseño sobre aspectos relevantes y evita una modelización excesivamente general.

2.2.3 Lenguaje de implementación

La metodología contempla la especificación del lenguaje formal con el que se implementará la ontología. En el caso de DataSecOnt, se ha optado por OWL (Web Ontology Language), sobre una base RDF (*Resource Description Framework*), lo que garantiza su compatibilidad con herramientas estándar y su capacidad para representar lógica descriptiva compleja.

2.2.4 Usuarios previstos

ORSD requiere la identificación explícita de los usuarios finales y técnicos que interactuarán con la ontología. Estos pueden incluir desarrolladores, investigadores, instituciones reguladoras o sistemas automáticos de consulta y razonamiento.

2.2.5 Usos previstos

Se definen los escenarios en los que se aplicará la ontología, ya sea para análisis semántico, consultas automáticas, visualización de conocimiento, clasificación de aplicaciones, evaluación de cumplimiento normativo, entre otros.

2.2.6 Requisitos ontológicos

La metodología distingue entre:

- **Requisitos funcionales**: expresados en forma de *Competency Questions* (CQs), que son preguntas en lenguaje natural que la ontología debe poder responder mediante consultas como SPARQL.
- **Requisitos no funcionales**: como el rendimiento, la interoperabilidad o la mantenibilidad del modelo, que aseguran su aplicabilidad técnica.

2.2.7 Agrupación de requisitos

Las preguntas de competencia se agrupan temáticamente para facilitar su análisis y cobertura durante el diseño del modelo. Esta agrupación puede orientarse a aspectos como el tipo de datos, los propósitos del tratamiento o las prácticas de seguridad.

2.2.8 Validación y priorización

Los requisitos son revisados y validados para garantizar su coherencia y viabilidad. Posteriormente, se asigna una prioridad a cada uno, en función de su relevancia para los objetivos del modelo.

2.2.9 Extracción de terminología

A partir de las preguntas y la documentación del dominio, se extrae un vocabulario inicial de términos clave, que sirve como base para construir la jerarquía conceptual, definir clases y relaciones, y asegurar la coherencia terminológica del modelo.

En este proyecto, la metodología ORSD ha sido empleada para definir los requisitos funcionales y no funcionales de la ontología DataSecOnt. La aplicación específica de cada tarea se detalla más adelante en el capítulo **¡Error! No se encuentra el origen de la referencia.**.

2.3 Justificación de la combinación metodológica

La metodología *Ontology Development 101* se ha elegido por su enfoque práctico, progresivo y ampliamente reconocido en el ámbito académico. Su carácter accesible permite aplicar un proceso sistemático de desarrollo de ontologías, incluso en proyectos que no requieren un alto grado de formalización inicial. Esta guía proporciona un conjunto de pasos claros que facilitan desde la delimitación del dominio hasta la creación de instancias y la validación de la ontología, fomentando la iteración y la mejora continua. Estas características la convierten en una base idónea para estructurar el diseño conceptual de la ontología y asegurar que todos los elementos esenciales queden recogidos de manera coherente [6].

La metodología *Ontology Requirements Specification Document* (ORSD) se ha adoptado como complemento por su capacidad para formalizar, de manera rigurosa, los requisitos que debe cumplir el modelo. ORSD aporta un marco detallado para documentar los objetivos, las preguntas de competencia, el alcance y los criterios de aceptación de la ontología. Este enfoque permite asegurar que el desarrollo no solo se base en decisiones técnicas, sino que responda a una especificación clara y alineada con las necesidades del dominio. Asimismo, su orientación documental facilita la trazabilidad y la futura reutilización del conocimiento generado [19].

La combinación de ORSD y *Ontology Development 101* permite afrontar el desarrollo de la ontología de manera completa, abarcando tanto la definición de los requisitos como su materialización en un modelo formal.

Por un lado, ORSD aporta una base documental clara que recoge qué se espera de la ontología y qué objetivos debe cumplir. Por otro, *Ontology Development 101* facilita un proceso ordenado para diseñar y estructurar los conceptos, relaciones y restricciones necesarias.

El uso combinado de ambas metodologías garantiza que el modelo resultante sea:

- Coherente a nivel semántico, porque se apoya en requisitos previamente definidos.
- Sólido desde el punto de vista técnico, al emplear estructuras y reglas bien establecidas en OWL.
- Flexible y mantenible, al quedar alineado con los objetivos del dominio y preparado para evolucionar si cambian las necesidades.

Esta aproximación dual ha permitido que la ontología DataSecOnt no solo represente de forma precisa la información declarada en Google Play sobre el uso de datos personales, sino que también sea capaz de evolucionar, adaptarse a nuevas preguntas y ser reutilizada en futuros trabajos académicos o técnicos.

2.4 Planificación

La planificación del desarrollo de la ontología DataSecOnt se ha basado en un modelo de ciclo de vida incremental, orientado a facilitar la evolución progresiva del modelo semántico y asegurar que cada fase del trabajo responda a los requisitos definidos. Este enfoque permite introducir mejoras de forma continua a partir de la validación, la retroalimentación recibida y la experiencia acumulada durante el proceso de desarrollo [21].

2.4.1 Enfoque adoptado

El ciclo de desarrollo se ha estructurado en iteraciones breves que integran distintas fases de análisis, diseño, implementación y evaluación. Este modelo ha favorecido la corrección temprana de errores, la adaptación a requisitos emergentes y la mejora progresiva de la calidad del modelo. Las etapas clave de cada iteración son:

1. **Análisis y Validación de Requisitos**: Cada iteración inicia con la revisión y ajuste de los requisitos, basándose en la metodología ORSD y los resultados de las iteraciones.
2. **Diseño y Refinamiento del Modelo**: Se desarrolla un modelo conceptual preliminar que incluye la definición de entidades, propiedades y relaciones siguiendo la metodología *Ontology Development 101*. Durante cada ciclo, se revisa y mejora el diseño en función de los resultados de las pruebas y de la validación de las preguntas de competencias mediante consultas SPARQL.
3. **Implementación y Validación**: La ontología se implementa utilizando OWL en herramientas como GraphDB. En cada iteración, se crean instancias de ejemplo y se ejecutan consultas SPARQL para validar que la ontología responde adecuadamente a las preguntas de competencia definidas.
4. **Documentación**: Al finalizar cada ciclo, se documentan los avances y se recopilan los resultados obtenidos que orientan las iteraciones futuras junto a una lista de tareas para la siguiente iteración.

2.4.2 Justificación del Enfoque

La adopción de un enfoque iterativo e incremental se justifica plenamente en el contexto de este proyecto, por varias razones [21]:

- **Flexibilidad:** Permite ajustar y refinar el modelo semántico en función de descubrimientos, cambios en los requisitos o comentarios recibidos durante el desarrollo, sin necesidad de rehacer por completo el trabajo anterior.
- **Validación continua:** Cada parte del modelo puede validarse individualmente mediante consultas y razonamiento lógico antes de ser integrada en el conjunto, garantizando así la consistencia global y la calidad del modelo final.
- **Reducción de riesgos:** Al detectar y resolver errores o inconsistencias de forma temprana, se minimizan los problemas en fases avanzadas del proyecto. Esto asegura una evolución más robusta y fiable del modelo ontológico.
- **Compatibilidad metodológica:** Este enfoque se integra de manera coherente con las metodologías *Ontology Development 101* y ORSD, ya que todas comparten un carácter iterativo que facilita la revisión constante y la mejora progresiva del modelo.

2.4.3 Descomposición del trabajo

Aunque la metodología principal seguida en este proyecto es ORSD para la especificación de requisitos ontológicos, se ha usado una descomposición del trabajo basada en el modelo WBS (*Work Breakdown Structure*) [22] como apoyo durante todo el proceso de desarrollo. Esta estructura permite representar no solo la aplicación del ORSD, sino también el conjunto de fases que abarca el ciclo completo del proyecto, incluyendo análisis de dominio, diseño ontológico, implementación técnica, validación y redacción final del TFG.

En la figura siguiente (Ilustración 2: WBS) se representan el diagrama completo del WBS, con todos los niveles de actividades.

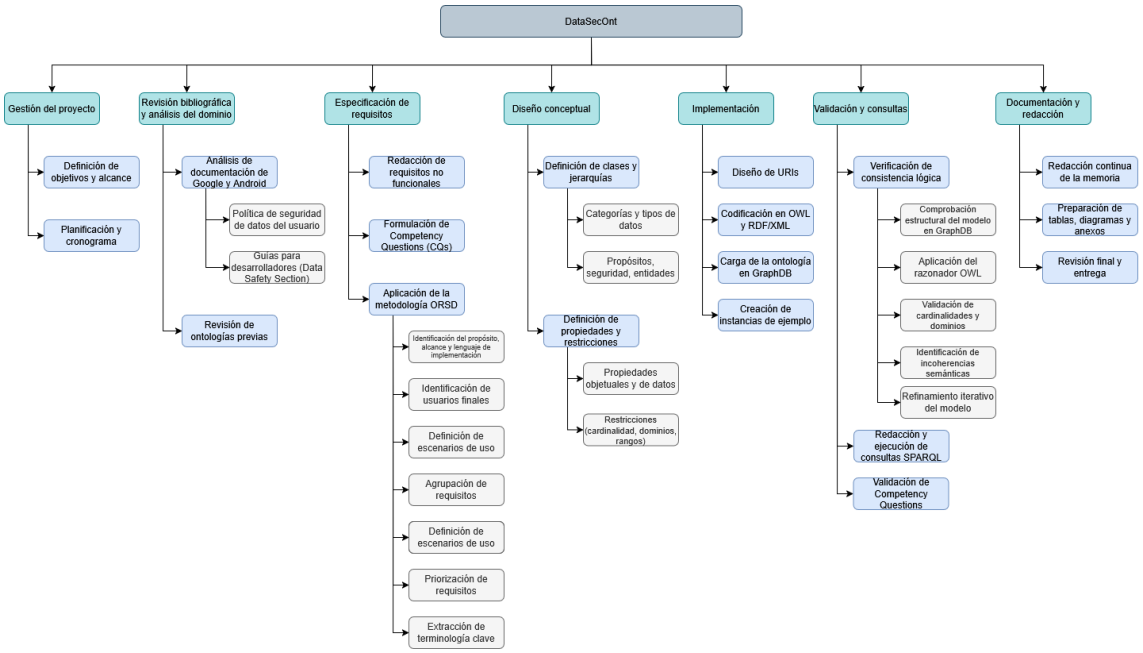

ILUSTRACIÓN 2: WBS

2.4.4 Cronograma

Con el fin de asegurar una planificación ordenada y coherente del proyecto DataSecOnt, se ha definido un cronograma detallado que establece las principales actividades a realizar a lo largo del periodo comprendido entre el 17 de febrero y finales de junio de 2025. Dicho cronograma está organizado por semanas y estructurado en función de las fases definidas en la descomposición del trabajo (WBS) [21, 22].

A lo largo del desarrollo del proyecto se han fijado una serie de hitos temporales clave, asociados al cierre de cada una de las etapas principales:

- Inicio del proyecto: 17 de febrero de 2025 (Semana 3 de febrero).
- Fin de la fase de gestión del proyecto: Semana 4 de febrero.
- Fin de la revisión bibliográfica y análisis del dominio: Semana 1 de marzo.
- Finalización de la especificación de requisitos (ORSD y CQs): Semana 1 de abril.
- Finalización del diseño conceptual: Semana 4 de abril.
- Inicio de la implementación técnica: Semana 4 de abril.
- Fin de la implementación: Semana 3 de mayo.
- Finalización de la validación de consultas y verificación de la ontología: Semana 1 o 2 de junio.

En la Ilustración 3: diagrama de gantt, se presenta el diagrama de Gantt, que permite visualizar la distribución temporal de las actividades, su duración estimada y sus dependencias. Como puede observarse, la documentación y redacción se ha planteado como una tarea transversal que acompaña todo el desarrollo del proyecto [21].

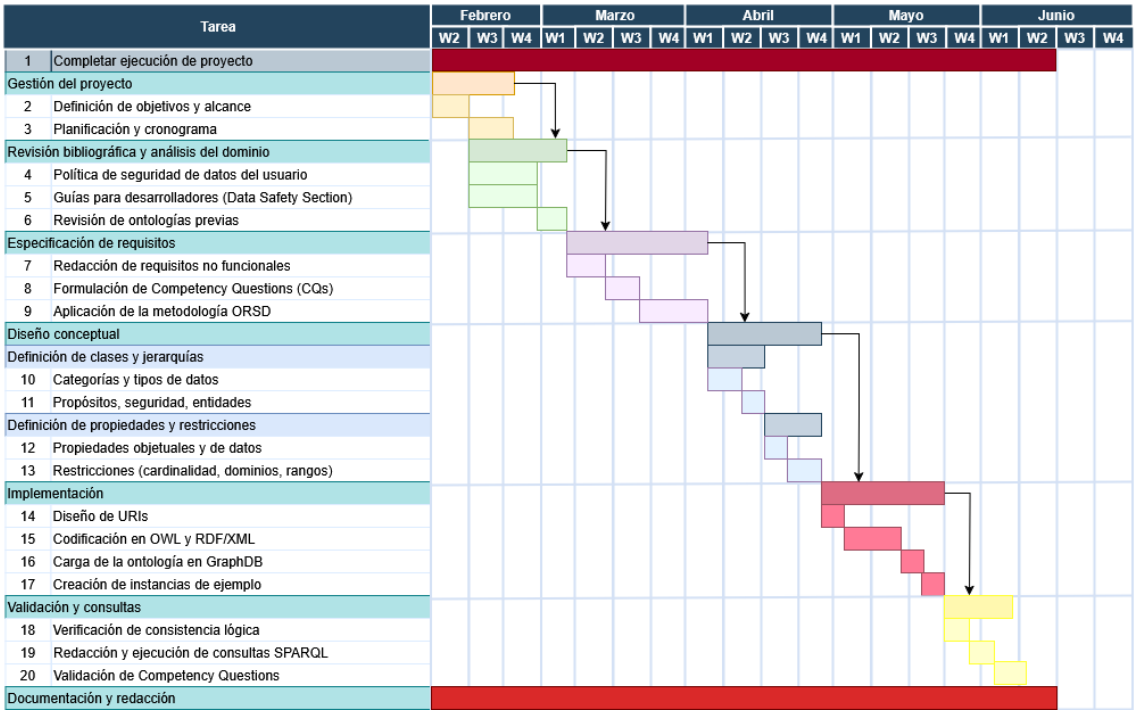

ILUSTRACIÓN 3: DIAGRAMA DE GANTT

2.4.5 Análisis de costes

Aunque este trabajo se ha desarrollado en el marco de un Trabajo de Fin de Grado, se ha estimado el coste del proyecto como si hubiera sido ejecutado por una profesional junior en condiciones reales de mercado. Esta estimación permite valorar el esfuerzo invertido y aproximar su viabilidad económica en un contexto profesional [21, 23].

El análisis se divide en diferentes tipos de coste: personal, indirectos, software, hardware y recursos cedidos por la universidad. A continuación, se detalla cada uno de ellos:

- **Coste de personal**: Se ha estimado una dedicación total de 250 horas por parte de la autora, a una tarifa aproximada de 12 €/hora, correspondiente al perfil de una ingeniera informática recién graduada. Esto da lugar a un coste total de 3.000 euros.
- **Costes indirectos**: Se consideran gastos asociados al uso de conexión a internet durante todo el periodo de trabajo. Se ha realizado una estimación aproximada de 50 euros para cubrir los 4 meses de trabajo.

- **Coste de software**: Todas las herramientas utilizadas (como GraphDB Free Edition y Escritorio Remoto de Google Chrome) son de libre acceso o gratuitas, por lo que el coste en licencias es nulo.
- **Infraestructura cedida**: El proyecto ha requerido el uso de una máquina virtual proporcionada por la Universidad de Valladolid, cuyo coste se ha estimado usando los datos para una máquina virtual usando Google Cloud. Se ha estimado mediante las herramientas de cálculo de presupuestos de Google Cloud, un uso mensual de 50 horas usando Ubuntu, siendo el coste mensual de 4.90 € al mes, lo que redondeando corresponde a 25 € para el proyecto.
- **Coste de hardware**: No se imputan costes por hardware, ya que el trabajo se ha realizado con equipo personal ya disponible (ordenador, teclado y ratón).

Tipo de coste	Concepto	Coste total
Personal	Ingeniero informático junior (250h x 12 €/h)	3000 €
Indirectos	Conexión a internet	50 €
Software	Licencias y herramientas	0 €
	Máquina virtual	25 €
Hardware	Ordenador, teclado, ratón (propios)	0 €
TOTAL ESTIMADO		3525 €

TABLA 1: ANÁLISIS DE COSTES

El proyecto DataSecOnt ha sido desarrollado al completo gracias al uso de herramientas gratuitas y recursos personales. El coste principal se concentra en el tiempo dedicado al trabajo, estimado en 3.000 euros, lo cual pone en valor el esfuerzo académico realizado.

2.4.6 Análisis de riesgos

Como parte del proceso de planificación, se ha llevado a cabo un análisis de riesgos con el objetivo de identificar posibles eventos que puedan afectar negativamente al desarrollo del proyecto DataSecOnt, evaluando tanto su probabilidad de ocurrencia como su impacto potencial sobre los resultados [21, 23].

Para ello, se ha utilizado una escala de valoración del riesgo basada en dos dimensiones:

- **Probabilidad**: Se mide en una escala de 1 a 5, donde 1 representa una ocurrencia muy poco probable y 5 representa una ocurrencia altamente probable.
- **Impacto**: Se mide también en una escala de 1 a 5, donde 1 representa un impacto mínimo y 5 un impacto crítico o severo.

La combinación de ambas dimensiones da lugar a un valor numérico denominado nivel de riesgo o daño, calculado como el producto de la probabilidad por el impacto. A partir de ese valor, se clasifica cada riesgo en una de las siguientes categorías:

1. Muy Bajo
2. Bajo
3. Medio
4. Alto
5. Muy Alto
6. Extremo

A continuación, se presenta la matriz de riesgos utilizada en este proyecto, donde se visualizan gráficamente los distintos niveles combinados de probabilidad e impacto:

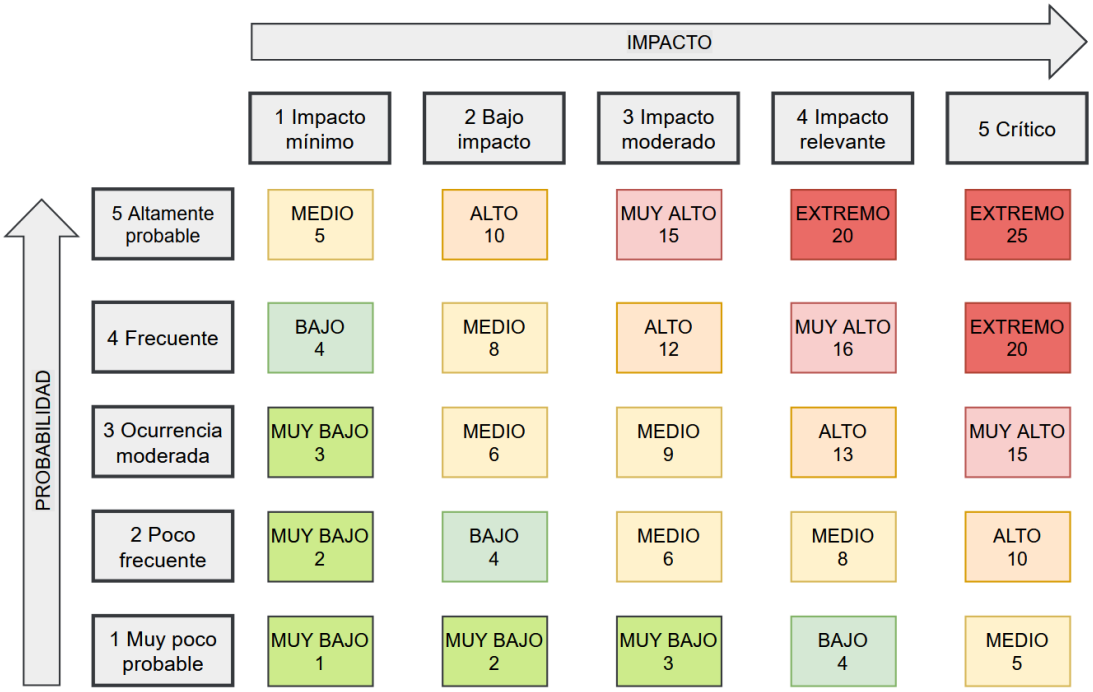

ILUSTRACIÓN 4: MATRIZ DE RIESGOS

Seguidamente, se muestra un listado detallado de los riesgos identificados, seguido de una tabla resumen en la que se recoge, para cada uno, su descripción, nivel de probabilidad, impacto estimado, nivel de daño, así como las medidas propuestas de prevención y mitigación.

1. Estimaciones optimistas o mala planificación
2. Recursos teóricos y conceptuales insuficientes
3. Uso de tecnologías desconocidas
4. Carga de trabajo adicional (académica o personal)
5. Imposibilidad de acceso al entorno de trabajo
6. Enfermedad o indisponibilidad del estudiante
7. Falta de apoyo o disponibilidad limitada de la tutora
8. El resultado final no cumple los objetivos definidos
9. Cambios o malentendidos en los requisitos
10. Pérdida de avances en el desarrollo
11. Errores de modelado semántico difíciles de detectar
12. Escasa motivación o agotamiento
13. Documentación incompleta o desactualizada
14. Dificultad en la validación automática
15. Cambios en la documentación oficial de Google Play

Riesgo 1	**Estimaciones optimistas o mala planificación**
Descripción	El cronograma inicial puede subestimar el tiempo real requerido para completar las tareas, generando acumulación de trabajo o retrasos
Probabilidad	3
Impacto	4
Daño	12 - Alto

Prevención	Establecer un cronograma realista con márgenes de seguridad
Mitigación	Reajustar tareas y prorrogar entregas parciales si es necesario

TABLA 2: RIESGO 1

Riesgo 2	**Recursos teóricos y conceptuales insuficientes**
Descripción	Dificultad para comprender o formalizar adecuadamente el dominio, especialmente al modelar la ontología
Probabilidad	3
Impacto	4
Daño	12 - Alto
Prevención	Ampliar la revisión bibliográfica desde el inicio.
Mitigación	Solicitar apoyo académico o bibliografía adicional a la tutora o investigadores.

TABLA 3: RIESGO 2

Riesgo 3	**Uso de tecnologías desconocidas**
Descripción	El uso de OWL, GraphDB y SPARQL puede conllevar una curva de aprendizaje que ralentice el progreso.
Probabilidad	3
Impacto	4
Daño	12 - Alto
Prevención	Realizar tutoriales y pruebas básicas al comienzo.
Mitigación	Buscar soporte técnico y adaptar herramientas si es necesario.

TABLA 4: RIESGO 3

Riesgo 4	**Carga de trabajo adicional (académica o personal)**
Descripción	La simultaneidad con otras responsabilidades puede interferir en el avance del TFG.
Probabilidad	5
Impacto	3
Daño	15 – Muy Alto
Prevención	Planificar anticipadamente fechas clave del calendario académico.
Mitigación	Reprogramar tareas menos prioritarias para liberar tiempo.

TABLA 5: RIESGO 4

Riesgo 5	**Imposibilidad de acceso al entorno de trabajo**
Descripción	Fallos técnicos, problemas con la máquina virtual o cortes de conexión pueden bloquear el trabajo.
Probabilidad	3
Impacto	3
Daño	9 – Medio
Prevención	Tener disponible software alternativo local o en la nube.
Mitigación	Cambiar de equipo o ubicación temporalmente.

TABLA 6: RIESGO 5

Riesgo 6	**Enfermedad o indisponibilidad del estudiante**
Descripción	Interrupciones personales por razones de salud u otras situaciones imprevistas.
Probabilidad	2
Impacto	4
Daño	8 – Medio
Prevención	Cuidar la salud y mantener reservas de tiempo.
Mitigación	Informar con antelación y solicitar ampliaciones de plazo.

TABLA 7: RIESGO 6

Riesgo 7	**Falta de apoyo o disponibilidad limitada de la tutora**
Descripción	Limitaciones de tiempo o coordinación que impidan validar avances importantes a tiempo.
Probabilidad	2
Impacto	3
Daño	6 – Medio
Prevención	Avanzar en el desarrollo del proyecto y consultar a la tutora en caso de dudas o problemas.
Mitigación	Consultar con otros miembros del departamento si es necesario.

TABLA 8: RIESGO 7

Riesgo 8	**El resultado final no cumple los objetivos definidos**
Descripción	El modelo ontológico puede no responder adecuadamente a las *Competency Questions*.
Probabilidad	3
Impacto	5
Daño	15 – Muy Alto
Prevención	Validar requisitos y CQs en cada iteración del diseño.
Mitigación	Ajustar el modelo y redefinir el alcance si es necesario.

TABLA 9: RIESGO 8

Riesgo 9	**Cambios o malentendidos en los requisitos**
Descripción	Una interpretación errónea o requisitos cambiantes pueden forzar rehacer partes del trabajo.
Probabilidad	3
Impacto	4
Daño	12 –Alto
Prevención	Aplicar las metodologías planteadas para recoger adecuadamente los requisitos.
Mitigación	Ajustar el modelo y redefinir el alcance si es necesario.

TABLA 10: RIESGO 9

Riesgo 10	**Pérdida de avances en el desarrollo**
Descripción	Pérdida de archivos, mal uso de versiones o fallos técnicos que eliminen progreso.
Probabilidad	2

Impacto	5
Daño	10 –Alto
Prevención	Usar control de versiones y copias de seguridad frecuentes.
Mitigación	Recuperar últimas versiones estables y verificar repositorios.

TABLA 11: RIESGO 10

Riesgo 11	**Errores de modelado semántico difíciles de detectar**
Descripción	Errores conceptuales en OWL o RDF que no se identifiquen hasta la fase de validación.
Probabilidad	4
Impacto	4
Daño	16 – Muy Alto
Prevención	Usar razonadores y validación intermedia frecuente.
Mitigación	Simplificar el modelo o modularizar las secciones conflictivas.

TABLA 12: RIESGO 11

Riesgo 12	**Escasa motivación o agotamiento**
Descripción	La extensión temporal del proyecto puede provocar fatiga o desánimo.
Probabilidad	3
Impacto	3
Daño	9 – Medio
Prevención	Fijar metas semanales claras y tiempos de descanso.
Mitigación	Replantear objetivos y dividir tareas grandes en subtareas manejables.

TABLA 13: RIESGO 12

Riesgo 13	**Documentación incompleta o desactualizada**
Descripción	Pérdida de coherencia en la memoria por documentación irregular durante el proceso.
Probabilidad	3
Impacto	3
Daño	9 – Medio
Prevención	Documentar avances en paralelo al desarrollo.
Mitigación	Usar el historial de versiones y comentarios para reconstruir decisiones.

TABLA 14: RIESGO 13

Riesgo 14	**Dificultad en la validación automática**
Descripción	Problemas al redactar y ejecutar consultas SPARQL que validen correctamente el modelo.
Probabilidad	3
Impacto	3
Daño	9 – Medio
Prevención	Probar cada consulta a medida que se define.
Mitigación	Dividir en subconsultas y depurar por partes.

TABLA 15: RIESGO 14

Riesgo 15	**Cambios en la documentación oficial de Google Play**
Descripción	Si Google modifica sus políticas o tipologías, puede afectar a la validez de la ontología.
Probabilidad	2
Impacto	4
Daño	8 – Medio
Prevención	Guardar versiones oficiales al comienzo del proyecto.
Mitigación	Adaptar la ontología a los cambios si son menores.

TABLA 16: RIESGO 15

Una vez que se han identificado y clasificado los riesgos, el paso siguiente es determinar la forma de gestionarlos. Habitualmente, las opciones disponibles son tres: transferir el riesgo a un tercero, evitar su aparición o reducir su impacto mediante acciones de mitigación [23]. En este proyecto se ha decidido mitigar aquellos riesgos cuya exposición se ha evaluado como alta o muy alta, mientras que los restantes se asumirán como parte de la ejecución normal del trabajo .

Capítulo 3 - Requisitos y análisis

3.1 Aplicación de *Ontology Development 101* a nuestro problema

3.1.1 Determinación de dominio y alcance

El primer paso en el desarrollo de la ontología DataSecOnt ha sido delimitar el dominio de conocimiento que se desea representar, en línea con la metodología *Ontology Development 101* [6]. Para ello, se han planteado las siguientes cuestiones clave:

- **¿Qué dominio cubrirá la ontología?**

  El dominio cubierto es el de la recopilación, tratamiento, gestión y protección de los datos personales y sensibles en aplicaciones móviles publicadas en Google Play, conforme a la información declarada en la sección de Seguridad de los Datos que Google exige a los desarrolladores [1].

- **¿Qué tipo de información debe representar y procesar?**

  La ontología debe representar el tipo de datos que puede recopilar una aplicación (ubicación, contactos, multimedia, etc.), su clasificación por categorías, los propósitos para los que son tratados, las condiciones en que se comparten (por ejemplo, con terceros), las prácticas de manejo (obligatorio/opcional), así como las medidas de seguridad aplicadas (encriptación en tránsito, mecanismos de eliminación, etc.). Además, debe indicar si los datos son considerados personales o sensibles según la documentación de Google [7].

- **¿Qué tipo de preguntas debe responder la ontología?**

  La ontología debe permitir formular y responder consultas sobre las prácticas de privacidad y seguridad de las aplicaciones, tales como:

  - ¿Qué datos sensibles recopila una aplicación determinada?
  - ¿Cuáles son los fines asociados a la recopilación de datos?
  - ¿Qué medidas de seguridad han sido declaradas?

  Esta pregunta está enlazada con las *Competency Questions* que se desarrollarán usando la metodología ORSD.

- **¿Quiénes serán los usuarios finales y responsables del mantenimiento?**

  Los usuarios finales previstos para esta ontología son principalmente grupos de investigación especializados en privacidad digital y análisis de aplicaciones móviles, como el grupo App-PI. Además, podrá ser empleada por académicos, desarrolladores y evaluadores interesados en realizar auditorías de aplicaciones o en estudiar el ecosistema de Google Play desde una perspectiva semántica. El mantenimiento, la actualización y la eventual ampliación de la ontología recaerán sobre estos mismos colectivos, que podrán adaptarla y reutilizarla en el marco de nuevas investigaciones o proyectos de desarrollo.

3.1.2 Consideración de ontologías existentes

Antes de proceder con la construcción de DataSecOnt desde cero, se evaluó la posibilidad de reutilizar ontologías ya existentes, en línea con las recomendaciones de la metodología *Ontology Development 101* [6]. Este paso tiene como objetivo promover la interoperabilidad semántica y aprovechar el conocimiento previamente.

Se tomó como referencia la ontología elaborada por el grupo App-PI, que modela los permisos solicitados por las aplicaciones en Google Play, incluyendo aspectos como los tipos de permiso, las categorías de riesgo y el acceso a funciones del dispositivo [13].

Aunque DataSecOnt se centra específicamente en la sección de Seguridad de los Datos de Google Play, y no en la gestión de permisos, se consideró que la ontología de App-PI [13] incluía conceptos y estructuras útiles que podían servir de referencia. Por este motivo, se reutilizaron ciertos patrones de modelado con el objetivo de mantener la coherencia conceptual y facilitar, en el futuro, una posible integración entre ambas ontologías.

3.1.3 Enumeración de términos relevantes

Se ha llevado a cabo una identificación exhaustiva de los términos más representativos del dominio, siguiendo las directrices propuestas en *Ontology Development 101* [6]. Esta fase constituye la base conceptual sobre la que se han construido posteriormente las clases, propiedades y restricciones de la ontología DataSecOnt.

El objetivo de este paso no es estructurar ni jerarquizar los conceptos, sino recoger de forma libre y sistemática todos aquellos términos que aparecen de forma recurrente en la documentación.

Para construir este listado, se ha respondido a las siguientes preguntas clave:

- ¿Qué términos o conceptos son necesarios para describir el dominio de forma precisa?
- ¿Qué atributos o propiedades caracterizan a estos términos?
- ¿Qué relaciones existen entre ellos y qué información aportan?

El vocabulario resultante incluye:

- **Sustantivos** clave como Aplicación, Tipo de Dato, Categoría de Dato, Propósito, Medida de Seguridad, Organización, Permiso, Instancia, Usuario, etc.
- **Verbos o relaciones** como recopila, comparte, gestiona, declara, implementa, requiere, procesa, etc.
- **Atributos** y valores relevantes como datoSensitivo, usoOpcional, procesamientoEfimero, requiereConsentimiento, encriptadoEnTransito, entre otros.

También se han considerado elementos específicos de las taxonomías impuestas por Google, tales como las categorías Ubicación, Información personal, Mensajes, Multimedia, Contactos, así como los propósitos declarativos: Funciones de la app, Análisis, Publicidad, Seguridad, etc [7].

Este conjunto inicial de términos no pretende definir aún la ontología final, pero constituye la materia prima del modelo semántico. A partir de esta recopilación, se irá depurando y ampliando en cada iteración del proceso de diseño.

3.1.4 Definición de clases y jerarquía

En el caso de la ontología DataSecOnt, se ha adoptado un enfoque combinando *top-down* y *bottom-up*, ya que permite una mayor flexibilidad respecto a la estructura conceptual del dominio.

Concretamente:

- Se han definido desde el inicio clases generales como Aplicación, TipoDeDato, Propósito, PrácticaDeSeguridad, que forman el esqueleto superior del modelo.

- Paralelamente, se han identificado clases específicas directamente extraídas de la documentación de Google Play (por ejemplo, UbicaciónPrecisa), que posteriormente se han integrado dentro de jerarquías semánticas apropiadas.

Este enfoque combinado ha resultado especialmente útil, ya que el dominio contiene categorías impuestas externamente, como las tipologías de datos y los propósitos de uso definidos por Google [7, 24], que era necesario reflejar de forma fiel. Además, algunos conceptos surgieron de manera emergente durante el análisis de casos concretos y a partir de la reorganización progresiva del vocabulario, lo que favoreció un planteamiento de construcción bottom-up. La validación iterativa mediante consultas SPARQL hizo necesario establecer jerarquías funcionales desde fases tempranas, con el fin de poder probar inferencias y relaciones de forma continua.

La definición concreta de las clases ontológicas, así como la asignación de propiedades (tanto de objeto como de datos), la especificación de sus características formales (dominios, rangos, cardinalidades o restricciones lógicas), y la creación de instancias representativas del dominio se desarrollan en detalle en el capítulo **¡Error! No se encuentra el origen de la referencia.**. Esta sección recoge la estructura formal del modelo DataSecOnt y muestra cómo se ha implementado cada elemento conforme a las metodologías previamente descritas.

3.2 Requisitos

Para garantizar un desarrollo sólido y alineado con las necesidades del dominio, la ontología DataSecOnt ha seguido la metodología *Ontology Requirements Specification Document* (ORSD) [19].

A continuación se describe paso por paso los puntos que define esta metodología:

3.2.1 Finalidad (*Purpose*)

El objetivo principal de la ontología DataSecOnt es modelar, representar y analizar de forma estructurada la información declarada en la sección de Seguridad de los Datos de Google Play, centrándose en el tratamiento de datos personales y sensibles por parte de aplicaciones móviles. La ontología busca facilitar la inferencia semántica, la clasificación de aplicaciones según sus prácticas de privacidad y el análisis mediante consultas SPARQL de sus políticas de uso y protección de datos.

3.2.2 Alcance (*Scope*)

La ontología cubre el dominio del tratamiento de datos en apps Android publicadas en Google Play, abarcando:

- Tipos de datos recopilados y su clasificación.
- Motivos y finalidades del uso de datos.
- Mecanismos de compartición y manejo.
- Prácticas de seguridad aplicadas.
- Cumplimiento de políticas y estándares relacionados.

Este conocimiento se estructura para permitir análisis automáticos y responder a preguntas relevantes para auditar el comportamiento de las apps respecto a la privacidad.

3.2.3 Lenguaje de implementación (*Implementation Language*)

La ontología está implementada utilizando OWL (*Web Ontology Language*), basado en RDF, lo que garantiza su compatibilidad con herramientas semánticas, motores de inferencia y lenguajes de consulta como SPARQL.

3.2.4 Usuarios finales previstos (*Intended End-Users*)

Como se ha comentado en la pregunta planteada por la metodología *Ontology Development 101*, los usuarios finales previstos son:

- Grupos de investigación sobre privacidad y análisis de datos (por ejemplo, App-Pi).
- Investigadores en Web Semántica, ontologías y protección de datos.
- Desarrolladores de herramientas de auditoría y clasificación de apps.
- Personas o empresas interesadas en auditar aplicaciones.
- Estudiantes y docentes en áreas de informática y datos personales.

3.2.5 Usos previstos (*Intended Uses*)

- Analizar las prácticas de manejo de datos por parte de aplicaciones móviles.
- Clasificar apps en función de los datos que manejan y cómo lo hacen.
- Realizar auditorías a apps dentro de un contexto de cumplimiento de la privacidad de datos.
- Validar el cumplimiento de buenas prácticas de privacidad.
- Responder a *Competency Questions* mediante consultas SPARQL.
- Apoyar estudios académicos sobre privacidad y semántica.

3.2.6 Requisitos de la ontología

3.2.6.1 Requisitos no funcionales

- **NFR1: Rendimiento**
  Las consultas SPARQL deben responder en menos de 500 ms en repositorios optimizados.
- **NFR2: Escalabilidad**
  La ontología debe admitir la incorporación de nuevas clases, propiedades y axiomas sin rediseño estructural.
- **NFR3: Mantenibilidad**
  Debe existir documentación clara y estructurada que facilite futuras actualizaciones.
- **NFR4: Compatibilidad**
  Debe cumplir con estándares del W3C como RDF, OWL y SPARQL.
- **NFR5: Usabilidad**
  La estructura debe ser comprensible, con ejemplos, anotaciones y consultas ilustrativas.
- **NFR6: Robustez**
  Debe garantizar la consistencia lógica del modelo y permitir su exportación en múltiples formatos (RDF/XML, Turtle, JSON-LD...).
- **NFR7: Accesibilidad**
  La ontología debe estar disponible mediante interfaces web o endpoints SPARQL públicos y cumplir principios básicos de accesibilidad.

3.2.6.2 Requisitos funcionales: Preguntas de Competencia (*Competency Questions*)

A continuación, se presentan las CQs formuladas para DataSecOnt:

1. **CQ1:** ¿Qué tipos de datos pueden recopilar las aplicaciones según la sección de Seguridad de los Datos?
2. **CQ2:** ¿Cómo se clasifican los datos recopilados por las aplicaciones?
3. **CQ3:** ¿Cómo se clasifican los datos compartidos por las aplicaciones?
4. **CQ4:** ¿Qué medidas de seguridad declaran implementar las aplicaciones?
5. **CQ5:** ¿Qué aplicaciones recopilan datos personales sensibles?
6. **CQ6:** ¿Qué aplicaciones recopilan datos personales y de qué tipo?
7. **CQ7:** ¿De qué formas se comparten los datos de las aplicaciones?
8. **CQ8:** ¿Cuáles son los distintos fines por los que se recopilan datos?
9. **CQ9:** ¿Cuáles son los distintos fines por los que se comparten datos?

10. **CQ10:** ¿Qué aplicaciones permiten la eliminación de datos?
11. **CQ11:** ¿Qué aplicaciones recopilan datos que son opcionales?
12. **CQ12:** ¿Qué tipos de datos se consideran sensibles según la ontología?
13. **CQ13:** ¿Qué tipos de datos tienen múltiples propósitos de uso?
14. **CQ14:** ¿Qué datos no se comparten con terceros?
15. **CQ15:** ¿Qué aplicaciones utilizan cifrado en tránsito como medida de seguridad?

3.2.6.3 Términos clave extraídos de las *Competency Questions*

Término	Frecuencia estimada	Frecuencia real
Tipo de dato/Dato	Alta	10
Aplicación	Alta	10
Recopilación	Alta	6
Clasificación / Categoría / Forma	Alta	3
Propósito / Finalidad	Alta	3
Compartición	Media	4
Sensibilidad	Media	2
Seguridad / Medidas de seguridad	Media	2
Eliminación de datos	Media	1
Cifrado	Baja	1
Opcional	Baja	1

TABLA 17: TÉRMINOS CLAVE

3.3 Identificación de las fuentes de información

Para la construcción de la ontología DataSecOnt, las fuentes consultadas se han centrado principalmente en documentación oficial de Google y Android.

A continuación, se detallan las principales fuentes utilizadas:

1. Documentación oficial de Google Play

El núcleo informativo del dominio se ha extraído de la documentación publicada por Google para desarrolladores en la plataforma Android Developers, concretamente en lo relativo a la sección de Seguridad de los Datos. Esta sección establece los requisitos que deben cumplir las aplicaciones en cuanto a la declaración del uso de datos personales de los usuarios.

Entre los documentos más relevantes consultados se encuentran:

- **Política de Datos del Usuario de Google Play**
  Define las obligaciones legales y éticas que deben asumir los desarrolladores respecto al tratamiento de datos personales, incluyendo aspectos como la recopilación, el uso compartido, la retención de datos y la gestión de consentimientos [24].
- **Guía para completar la sección de Seguridad de los Datos**
  Documento destinado a los desarrolladores que describe, de forma detallada, cómo deben declarar el tratamiento de datos en sus aplicaciones. Incluye la tipología de datos admitidos, las categorías semánticas propuestas por Google, y los propósitos permitidos para su uso y compartición [7].

- **Documentación técnica sobre prácticas de seguridad y verificación (MASVS)**
  Establece criterios técnicos sobre medidas de seguridad recomendadas para las aplicaciones, tales como cifrado de datos en tránsito, validación de acceso o eliminación de datos a solicitud del usuario [25].

Esta documentación ha servido como base para identificar los conceptos, relaciones y restricciones que conforman el dominio y que posteriormente han sido representados en la ontología.

3 Ontologías y modelos preexistentes

Además de las fuentes oficiales, se ha considerado el trabajo desarrollado por el equipo de investigación App-PI [13], que se centra en el análisis semántico de los permisos en aplicaciones Android. La ontología generada por este grupo ofrece una representación preliminar del ecosistema de datos en Google Play [7], principalmente orientada a la gestión de permisos y políticas de acceso. Asimismo, las ontologías PPO [12] y DPV [11] han servido como referencia a la hora de definir el modelo OWL en formato RDF utilizado en este proyecto.

4 Recursos adicionales

Durante el proceso de desarrollo también se han consultado artículos académicos, entradas de la base de datos Google Play Console Help [26], y documentación técnica de iniciativas como OWASP MASVS (Mobile Application Security Verification Standard) [25], que definen prácticas de seguridad para aplicaciones móviles.

3.4 Definiciones conceptuales básicas

En este apartado se recogen las definiciones clave que fundamentan el desarrollo de la ontología DataSecOnt, en el contexto del tratamiento de datos personales en aplicaciones móviles disponibles en Google Play. Estas definiciones permiten estructurar formalmente los conceptos más relevantes relacionados con la recopilación, uso, manejo, protección y compartición de datos, así como con las prácticas de seguridad declaradas por los desarrolladores [7].

1. Recopilación y compartición de datos

- **Datos recopilados**: Información que la aplicación obtiene directamente del usuario (mediante formularios o permisos) o de forma automática a través de sensores, APIs u otros mecanismos.
- **Datos compartidos**: Información que, una vez recopilada, se transfiere a terceros. Esta transferencia puede realizarse:
  - A servidores externos (fuera del dispositivo).
  - A otras aplicaciones en el mismo dispositivo.
  - A través de bibliotecas de terceros o SDKs integrados.
  - Mediante componentes WebView incrustados en la app.
- **Organización de origen**: Entidad responsable del procesamiento de los datos recopilados por la aplicación.
- **Organización de terceros**: Cualquier entidad ajena a la organización de origen o a sus proveedores directos que intervenga en la recopilación o tratamiento de los datos.

2. Procesamiento y manejo de los datos

- **Procesamiento efímero**: Indica si los datos se tratan de forma transitoria y sin almacenamiento persistente (verdadero) o si permanecen guardados durante periodos prolongados (falso).
- **Manejo de datos**:
    - **Opcional**: El usuario puede decidir si habilita la recopilación de determinados datos.
    - **Obligatorio**: El funcionamiento de la aplicación requiere necesariamente la recopilación de esos datos.

3. Propósitos del tratamiento de datos

Las aplicaciones deben declarar los fines para los que recopilan o comparten información del usuario. Estos propósitos se agrupan en dos categorías principales:

- **Propósitos de recopilación**:
    - **Administración de la cuenta**: Configuración o gestión de la cuenta del usuario con el desarrollador.
    - **Publicidad o marketing**: Presentación o segmentación de anuncios y comunicaciones comerciales, así como medición del rendimiento publicitario.
    - **Funciones de la app**: Provisión de funcionalidades disponibles en la aplicación.
    - **Análisis**: Obtención de datos sobre el uso y el rendimiento de la aplicación.
    - **Comunicaciones del desarrollador**: Envío de notificaciones o mensajes relacionados con la app o el desarrollador.
    - **Seguridad, cumplimiento y prevención de fraudes**: Garantizar la seguridad e integridad de la aplicación, cumplir obligaciones legales y detectar o prevenir fraudes, abusos o usos indebidos.
    - **Personalización**: Adaptación de la aplicación al usuario, incluyendo contenidos o recomendaciones personalizadas.
- **Propósitos de compartición**:
    - **A un tercero**: Transferencia de datos a un tercero como consecuencia de una acción iniciada por el usuario y esperada de forma razonable.
    - **A un proveedor**: Transferencia de datos a un proveedor que procesa la información en nombre del desarrollador.
    - **Gubernamental**: Transferencia con fines legales específicos, por ejemplo, en respuesta a una solicitud de autoridades competentes.
    - **Anonimización**: Transferencia tras un proceso que elimina toda posibilidad de identificación personal.

4. Prácticas de seguridad y privacidad

Prácticas que las aplicaciones pueden declarar para proteger la información de los usuarios:

- **Encriptación en tránsito**: Uso de cifrado durante la transferencia de datos entre el dispositivo y los servidores.
- **Mecanismo de solicitud de eliminación**: Posibilidad de que el usuario solicite la supresión de sus datos personales, conforme a regulaciones como el Reglamento General de Protección de Datos (RGPD).

5. Insignias de cumplimiento normativo

- **Política de Familias (Google Play)**: Declaración del cumplimiento de esta política en aplicaciones dirigidas a menores, visible mediante una insignia específica.
- **Revisión de seguridad independiente**: Evaluación de terceros basada en estándares reconocidos, como MASVS de OWASP.
- ***Unified Payments Interface*** **(UPI)**: Sistema estandarizado de pagos en India, cuya compatibilidad puede declararse como garantía de seguridad adicional.

6. Tipología de datos

La ontología contempla un conjunto amplio de categorías de datos, organizadas semánticamente. A continuación, se describen las principales:

- **Ubicación**
    - **Ubicación aproximada**: Corresponde a la localización física del dispositivo en un área de al menos 3 kilómetros cuadrados, por ejemplo, la ciudad en la que se encuentra.
    - **Ubicación precisa**: Hace referencia a la localización física en un área inferior a 3 kilómetros cuadrados.
- **Información personal**:
    - **Nombre**: Forma de identificación, como nombre, apellido o sobrenombre.
    - **Dirección de correo electrónico**: Correo electrónico asociado al usuario.
    - **ID de usuario**: Identificadores vinculados a una persona identificable, como un número de cuenta o un identificador único.
    - **Dirección**: Domicilio particular o dirección postal.
    - **Número de teléfono**: Número telefónico de contacto.
    - **Raza y etnia**: Información declarada sobre raza o etnia.
    - **Creencias políticas o religiosas**: Información sobre creencias políticas o religiosas.
    - **Orientación sexual**: Datos relativos a la orientación sexual.
    - **Otra información**: Otros datos personales relevantes, como fecha de nacimiento, identidad de género o condición de veterano.
- **Información financiera**:

    - **Información de pago del usuario**: Datos sobre cuentas financieras, incluidos números de tarjeta de crédito.
    - **Historial de compras**: Información sobre transacciones realizadas.
    - **Calificación crediticia**: Datos relacionados con historial o calificación crediticia.
    - **Otra información financiera**: Información adicional de carácter financiero, como ingresos o deudas.
- **Salud y actividad física**:
    - **Información sanitaria**: Datos sobre el estado de salud, incluidos historiales clínicos o síntomas.
    - **Información de estado físico**: Información sobre actividad física o condición física general.
- **Mensajes**:
    - **Correos electrónicos**: Correos electrónicos, incluidos asunto, remitente, destinatario y contenido.
    - **SMS o MMS**: Mensajes de texto, con detalles de remitente, destinatario y contenido.
    - **Otros mensajes desde apps**: Mensajes instantáneos u otro contenido de chat.
- **Multimedia**:
    - **Fotos**: Imágenes almacenadas o generadas.
    - **Videos**: Archivos de vídeo almacenados o generados
    - **Audio**
        - **Grabaciones de voz o sonido**: Archivos de voz o sonido grabados.
        - **Archivos de música**: Música almacenada en el dispositivo.
        - **Otros archivos de audio**: Cualquier otro tipo de contenido sonoro.
- **Archivos y documentos**: Archivos o documentos generados o almacenados, incluida información como nombres de archivo.
- **Calendario**:
    - **Eventos del calendario**: Información sobre eventos, notas asociadas y asistentes.
- **Contactos**: Datos relacionados con los contactos, como nombres, historial de mensajes, grafos sociales, identificadores, antigüedad, frecuencia de contacto, duración de las interacciones y registro de llamadas.
- **Actividad en apps**:
    - **Interacciones en la app**: Información sobre el uso de la aplicación, por ejemplo, visitas de páginas o elementos seleccionados.

    - **Historial de búsqueda en la app**: Registros de búsquedas realizadas dentro de la aplicación.
    - **Apps instaladas**: Lista de aplicaciones instaladas en el dispositivo.
    - **Otro contenido generado por usuarios**: Contenido que no encaja en otras categorías, como biografías o respuestas abiertas.
    - **Otras acciones**: Actividades diversas, como interacciones en juegos, marcadores o elecciones en diálogos.
- **Navegación web**:
    - **Historial de navegación web**: Registros de sitios web visitados.
- **Información y rendimiento de la app**:
    - **Registros de fallas**: Datos sobre incidencias o errores de la aplicación.
    - **Diagnóstico**: Información técnica sobre rendimiento, como duración de batería, tiempos de carga o velocidad de fotogramas.
    - **Otros datos de rendimiento de apps**: Cualquier otro indicador de funcionamiento no recogido en categorías anteriores.
- **Identificadores de dispositivo**:
    - **Dispositivo u otros ID**: Códigos asociados a un dispositivo, navegador o aplicación, tales como número IMEI, dirección MAC, ID de dispositivo Widevine, identificador de instalación de Firebase o identificador publicitario.

Se incluye en el Anexo AA.1 Tabla la tipología de datos estructurada en tabla para facilitar la consulta

Capítulo 4 - Diseño conceptual

El diseño conceptual de la ontología DataSecOnt parte del análisis de los conceptos clave identificados en la documentación de Google Play sobre la sección de Seguridad de los Datos [7].

A partir de esta información, se han definido las clases principales, sus relaciones jerárquicas y propiedades más relevantes, con el objetivo de representar de forma estructurada la información que las aplicaciones deben proporcionar sobre el tratamiento de los datos de los usuarios.

4.1 Clases y jerarquía

Las clases principales de la ontología son:

- Aplicación
- TipoDeDato
- CategoriaDeDato
- Propósito
- Comparticion
- Manejo
- Seguridad
- Entidad

Estas clases se organizan jerárquicamente mediante relaciones semánticas como rdfs:subClassOf. Se han definido subclases como DatoPersonal, DatoFinanciero, DatoUbicacion, entre otras, que permiten una mayor granularidad en las consultas y razonamientos automáticos.

Esta estructura jerárquica favorece la inferencia lógica. Por ejemplo, si un TipoDeDato pertenece a la categoría DatoPersonal, se pueden aplicar automáticamente restricciones o reglas específicas asociadas a dicha categoría, como tratamiento como dato sensible o eliminación obligatoria.

Estas subclases que hemos definido iterativamente usando las metodologías a partir de la información del apartado 3.4 Definiciones conceptuales básicas, enriquecen el modelo y permiten representar diferentes aspectos del ciclo de vida del dato, así como las entidades que interactúan con ellos, diferenciando entre la organización responsable de la aplicación y terceras partes a quienes se puede transferir información.

Se presenta un diagrama representativo en la Ilustración 10: tipo de dato (primera parte), en el Anexo A sección A.2 Diagramas para tipo de dato.

4.2 Propiedades y relaciones

Durante el diseño conceptual de la ontología, se identificaron un conjunto de relaciones clave necesarias para representar el dominio de conocimiento. Estas relaciones permiten reflejar cómo las aplicaciones móviles gestionan los datos, con qué finalidades, qué entidades participan en su tratamiento y qué medidas de protección se aplican. Entre ellas se definen:

- La relación entre cada aplicación y los tipos de datos que declara recopilar.
- La relación entre los datos y los propósitos de recopilación y compartición.

- Las conexiones entre aplicaciones y las entidades externas con las que comparten información.
- El vínculo entre cada aplicación y las prácticas de seguridad que implementa, como el uso de cifrado.
- La indicación de si ciertos datos son considerados sensibles u opcionales.
- La descripción de las formas específicas en que se produce la compartición de datos (por ejemplo, con proveedores o con fines legales).

La definición de las propiedades y relaciones de la ontología se ha realizado en coherencia con las metodologías *Ontology Development 101* [6] y ORSD [19] aplicadas en este proyecto.

En particular, las fases de enumeración de términos, y la determinación del dominio y alcance, permitieron recopilar de manera exhaustiva los conceptos clave y las interacciones que debían modelarse. Por otro lado, ORSD [19] ayudó a alinear estas relaciones con los requisitos de información y las preguntas de competencia definidas inicialmente.

De este modo, se garantizó que cada relación conceptual respondiera a una necesidad concreta del dominio y contribuyera a la capacidad de la ontología para resolver consultas semánticas relevantes.

4.2.1 Características de las propiedades

Para garantizar la coherencia lógica y la expresividad de la ontología, se han definido distintas características sobre las propiedades:

- **Cardinalidad:**

  Algunas propiedades, como recopilaDato, comparteCon o tieneProposito, permiten múltiples valores, ya que una aplicación puede recopilar diversos tipos de datos, compartirlos con varias entidades o declararlos para distintos fines. En cambio, propiedades como procesamientoEfimero o permiteEliminacionDatos tienen cardinalidad uno, dado que representan valores booleanos únicos por cada instancia.

- **Tipo de valor:**

  Se han contemplado diferentes tipos de valores según la naturaleza de la información:

  - Propiedades booleanas, como procesamientoEfimero, datoEsSensitivo o permiteEliminacionDatos.
  - Propiedades de tipo cadena de texto, como modoComparticion y tipoManejo.
  - Propiedades de objeto, como recopilaDato o tieneProposito, que definen relaciones entre instancias de clases de la ontología.

- **Dominio y rango:**

  Cada propiedad se ha definido con un dominio y un rango explícitos, con el objetivo de asegurar que las relaciones sean semánticamente válidas y evitar ambigüedades durante la inferencia y la validación automática.

- **Obligatoriedad:**

  Determinadas propiedades, como recopilaDato, tieneProposito o permiteEliminacionDatos, se consideran obligatorias para las instancias de la clase Aplicacion, ya que contienen información fundamental sobre su comportamiento respecto a los datos del usuario.

- **Restricciones semánticas adicionales:**

Se han incluido axiomas OWL que establecen, por ejemplo, disyunciones entre subclases como DatoPersonal y DatoFinanciero, restricciones de valor que permiten inferir automáticamente si un TipoDeDato es sensible, así como reglas que clasifican un dato según su propósito declarado.

Capítulo 5 – Implementación

5.1 Diseño de URIs

El diseño de los identificadores uniformes de recurso (URIs) es un aspecto fundamental para asegurar la claridad, la coherencia y la interoperabilidad de la ontología. Todas las URIs empleadas en DataSecOnt pueden consultarse en detalle en el Anexo B – URIs.

Como punto de partida, se ha definido el espacio de nombres principal (namespace) de la ontología:

```
@prefix ds: <http://tfg.uva.es/ontologias/datasecont#>.
```

Siguiendo las recomendaciones y buenas prácticas en la construcción de URIs, se han aplicado las siguientes pautas [27]:

- Utilizar CamelCase para los nombres de clases (TipoDeDato, Aplicacion, Proposito) y snake_case en casos concretos si mejora la legibilidad.
- Emplear verbos o expresiones verbales en minúsculas para las propiedades, con nombres claros que reflejen su función (recopilaDato, comparteCon, tieneProposito).
- Evitar espacios, tildes, caracteres especiales o mayúsculas innecesarias que puedan dificultar la interpretación automática.
- Mantener la consistencia en la nomenclatura, de forma que todas las entidades se definan de manera homogénea.

La forma recomendada y utilizada en el modelo es la siguiente:

```
http://tfg.uva.es/ontologias/datasecont#NombreEntidad
```

Por ejemplo:

- http://tfg.uva.es/ontologias/datasecont#DatoPersonal
- http://tfg.uva.es/ontologias/datasecont#recopilaDato

Es importante destacar que, aunque podría parecer válido construir URIs con rutas jerárquicas intermedias, como:

```
http://tfg.uva.es/ontologias/datasecont/DatoPersonal/#Nombre
```

en RDF y OWL el símbolo # ya actúa como delimitador del fragment identifier, indicando que todo lo que figura a continuación es el identificador de un recurso dentro del documento raíz.

La jerarquía conceptual (por ejemplo, que DatoPersonal sea una subclase de TipoDeDato) no depende de que la URI contenga rutas anidadas o carpetas, sino que se define explícitamente mediante axiomas OWL y propiedades como rdfs:subClassOf o rdf:type. Por este motivo, se ha optado por un esquema de URIs plano y consistente, en el que el namespace base y el fragmento identificador son suficientes para localizar y describir cada entidad semántica del modelo.

Este diseño garantiza la claridad, facilita la lectura, y asegura que la ontología pueda ser interpretada por herramientas estándar de la Web Semántica de forma correcta.

5.2 Esquema RDF/XML

A continuación, se presenta la implementación formal de la ontología DataSecOnt utilizando el formato RDF/XML. Esta representación permite describir de forma estructurada y semánticamente precisa los conceptos del dominio, sus relaciones, propiedades y restricciones [14].

La ontología ha sido desarrollada conforme a las recomendaciones de OWL (*Web Ontology Language*), incluyendo la definición de clases, subclases, propiedades de objeto, propiedades de datos, dominios, rangos, y axiomas básicos como jerarquías o restricciones. El uso de RDF/XML garantiza la compatibilidad con herramientas de razonamiento, visualización y consulta como *Protégé*, GraphDB o motores SPARQL.

El contenido del fichero que se muestra a continuación refleja la versión actual del modelo ontológico, incorporando todos los elementos definidos en el diseño conceptual, el diseño de URIs y las jerarquías semánticas detalladas en las secciones anteriores.

5.3 Declaración formal de la ontología

La implementación completa de la ontología DataSecOnt se ha desarrollado utilizando OWL en formato RDF/XML, conforme a los estándares del W3C [28, 29]. Este documento describe de manera formal todas las clases, propiedades, restricciones y axiomas que componen el modelo.

Dado que el contenido completo resulta extenso, se ha incluido el enlace al repositorio que incluye la versión íntegra de la ontología en el Anexo C – Código del TFG, donde puede consultarse en detalle. A continuación, se presenta un fragmento representativo que ilustra la estructura general del RDF/XML empleado

```
    <rdf:RDF xmlns="http://tfg.uva.es/ontologias/datasecont#"
        xml:base="http://tfg.uva.es/ontologias/datasecont"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

      <owl:Ontology rdf:about="http://tfg.uva.es/ontologias/datasecont">
       <rdfs:label>DataSecOnt</rdfs:label>
       <rdfs:comment>Ontología sobre seguridad y privacidad de datos en Google
Play</rdfs:comment>
       <owl:versionInfo>v1.0</owl:versionInfo>
      </owl:Ontology>

      <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#Aplicacion"/>
      <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#TipoDeDato"/>
      <owl:ObjectProperty
rdf:about="http://tfg.uva.es/ontologias/datasecont#recopilaDato">
       <rdfs:domain rdf:resource="http://tfg.uva.es/ontologias/datasecont#Aplicacion"/>
       <rdfs:range rdf:resource="http://tfg.uva.es/ontologias/datasecont#TipoDeDato"/>
      </owl:ObjectProperty>

      <!-- Otros elementos se encuentran descritos en el anexo -->
    </rdf:RDF>
```

Este fragmento inicial declara la ontología DataSecOnt e incorpora los espacios de nombres estándar necesarios para garantizar la compatibilidad con RDF, RDFS y OWL. La etiqueta <owl:Ontology>

identifica el documento como una ontología OWL y permite incluir metadatos relevantes, como el nombre, la descripción general y la versión del modelo.

A continuación, se definen dos clases principales (Aplicacion y TipoDeDato), junto con una propiedad de objeto (recopilaDato) que establece la relación entre una aplicación y los tipos de datos que puede recopilar. Las propiedades rdfs:domain y rdfs:range determinan los dominios y rangos de esta relación, asegurando que las instancias sean semánticamente coherentes y puedan ser procesadas por motores de razonamiento.

5.4 Estructura interna

5.4.1 Clases principales

A continuación, se muestra un fragmento de la definición RDF/XML que recoge las clases principales de la ontología DataSecOnt. En este resumen se incluyen tanto las clases fundamentales de la jerarquía TipoDeDato (que agrupa las distintas categorías de datos susceptibles de recopilación y tratamiento), como otras clases relacionadas con el modelo general, tales como las aplicaciones, los propósitos declarados, las entidades involucradas y las prácticas de seguridad.

```
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#Aplicacion">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#TipoDeDato">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoUbicacion">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoPersonal">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoFinanciero">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoSalud">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoMensaje">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoMultimedia">
    <owl:Class
rdf:about="http://tfg.uva.es/ontologias/datasecont#ArchivosYDocumentos">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoCalendario">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoContacto">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoActividadApp">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoNavegacionWeb">
    <owl:Class
rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoRendimientoApp">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#DatoDispositivo">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#Proposito">
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#Entidad" />
    <owl:Class rdf:about="http://tfg.uva.es/ontologias/datasecont#Seguridad" />
```

Estas clases constituyen la base conceptual sobre la que se definen las propiedades, relaciones y restricciones semánticas de la ontología. El código completo para la versión final de la ontología puede consultarse en el Anexo C – Código del TFG.

5.4.2 Propiedades

Propiedades de objeto

A continuación se describen las principales propiedades de objeto definidas en la ontología. Estas propiedades permiten modelar las relaciones entre entidades clave, como aplicaciones, tipos de datos, propósitos de uso y medidas de seguridad.

- **recopilaDato**
    - **Dominio:** Aplicacion
    - **Rango:** TipoDeDato

- **Descripción:** Relaciona una aplicación con los tipos de datos que declara recoger. Es la propiedad central que permite identificar qué información personal o técnica recopila cada instancia de Aplicacion.

- **comparteCon**
  - **Dominio:** Aplicacion
  - **Rango:** Entidad
  - **Descripción:** Indica con qué entidad o tercero se comparten los datos recopilados. Permite reflejar transferencias de información a organizaciones externas o proveedores de servicios.

- **esRequeridoPara**
  - **Dominio:** TipoDeDato
  - **Rango:** PropositoRecopilacion
  - **Descripción:** Especifica que un determinado tipo de dato es necesario para alcanzar un propósito concreto de recopilación. Facilita documentar la justificación funcional de la recogida de datos.

- **seCompartePara**
  - **Dominio:** TipoDeDato
  - **Rango:** PropositoComparticion
  - **Descripción:** Relaciona el tipo de dato con el propósito por el que se comparte con terceros, identificando la finalidad declarada de dicha transferencia.

- **usaEncriptacion**
  - **Dominio:** Aplicacion
  - **Rango:** Seguridad
  - **Descripción:** Indica si la aplicación aplica medidas de cifrado en tránsito o almacenamiento como mecanismo de protección de la información.

- **implementaMedidaSeguridad**
  - **Dominio:** Aplicacion
  - **Rango:** Seguridad
  - **Descripción:** Permite detallar qué medidas de seguridad adicionales se aplican en la aplicación, más allá de la encriptación.

- **esPropositoDeComparticion**
  - **Dominio:** Proposito
  - **Rango:** PropositoComparticion

- **Descripción:** Relaciona un propósito genérico con su rol específico en la compartición de datos. Esta propiedad facilita clasificar propósitos en función de si motivan la transferencia de información a terceros.

- **esPropositoDeRecopilacion**
  - **Dominio:** Proposito
  - **Rango:** PropositoRecopilacion
  - **Descripción:** Relaciona un propósito genérico con su papel en la recopilación de datos. Permite modelar la finalidad de cada recogida declarada en la sección de Seguridad de los Datos.

Cabe destacar, que en la ontología DataSecOnt, las propiedades esPropositoDeRecopilacion y esPropositoDeComparticion se utilizan para caracterizar conceptualmente los tipos de propósito. Es decir, estas relaciones permiten clasificar cada instancia de propósito como perteneciente a una de las dos grandes categorías: recopilación de datos o compartición de datos. Por ejemplo, Publicidad o Marketing es un propósito que se vincula mediante esPropositoDeComparticion a la clase PropositoComparticion.

En cambio, las propiedades esRequeridoPara y seCompartePara expresan relaciones más específicas entre un TipoDeDato concreto y el propósito que motiva su tratamiento. Así, esRequeridoPara indica que un dato se recopila con un fin determinado (por ejemplo, la recogida de un correo electrónico requerida para la funcionalidad de la aplicación), mientras que seCompartePara señala que el dato se comparte con un fin específico (como compartir un identificador de usuario para fines publicitarios).

En otras palabras, esPropositoDeRecopilacion y esPropositoDeComparticion clasifican el propósito en el modelo conceptual, mientras que esRequeridoPara y seCompartePara describen las relaciones prácticas entre datos concretos y los usos que justifican su recogida o su cesión a terceros.

5.4.3 Ejemplos de instancias

A continuación se describen algunos casos reales de aplicaciones móviles a partir de un JSON obtenido con web scraping sobre Google Play, que se han aplicado a la ontología, resaltando los datos recogidos, sus propósitos de uso y medidas de seguridad implementadas.

Aplicación Llaollao (com.llaollao.app)

Es la plataforma oficial de la cadena de yogurt helado natural llaollao, creada para digitalizar y mejorar la experiencia de fidelización y comunicación con sus clientes [30].

Datos recopilados

Recoge un conjunto limitado de datos personales y de salud:

- Información física (opcional), como datos de fitness.
- Identificadores de usuario: nombre, correo electrónico, número de teléfono e identificador de dispositivo (todos obligatorios).

Propósitos: Los datos se utilizan exclusivamente para el funcionamiento básico de la aplicación y la gestión de las funcionalidades asociadas al perfil de usuario.

Medidas de seguridad: Se garantiza la posibilidad de eliminación por parte del usuario y se implementa cifrado en tránsito.

Aplicación Bonpreu (com.bonpreu.mobile.android)

Aplicación de comercio electrónico del Grup Bon Preu, uno de los principales distribuidores de alimentación en la comunidad autónoma de Cataluña y está diseñada para facilitar y mejorar la experiencia de compra tanto en los supermercados físicos Bonpreu y Esclat como en su tienda online [31].

Datos recopilados

Recoge:

- Información sobre interacciones en la app y búsquedas.
- Historial de compras y datos de pago (obligatorios).
- Identificadores (nombre y User ID).
- Información técnica como registros de fallos y diagnósticos.

Propósitos: Parte de la información (historial de compras e identificadores) se comparte con fines de publicidad y marketing, mientras que el resto se utiliza para análisis de uso y funcionamiento interno.

Medidas de seguridad: Admite la eliminación de datos y emplea cifrado en tránsito.

Aplicación Pinterest (com.pinterest)

Red social visual que permite a los usuarios crear, administrar y compartir colecciones de imágenes y vídeos organizados en tableros temáticos. Estos tableros funcionan como murales digitales donde se "*pinean*" (fijan) imágenes relacionadas con intereses, eventos, hobbies o cualquier tema que el usuario desee [32].

Datos recopilados

Incluye:

- Ubicación precisa (opcional).
- Historial de compras, historial de búsqueda y User ID (compartidos con fines de marketing).
- Registros de fallos.

Propósitos: La información se usa principalmente para análisis de comportamiento, personalización del contenido y campañas publicitarias.

Medidas de seguridad: Permite la eliminación de datos y aplica cifrado en tránsito.

Aplicación SaraMart (com.saramart.android)

La aplicación SaraMart, actualmente rebautizada como Hacoo, es un marketplace de origen chino que ofrece productos de moda, tecnología y hogar a precios muy bajos, lo que la ha hecho especialmente popular entre la generación Z y en redes sociales como TikTok. Nació como alternativa a AliExpress y Temu, con un catálogo repleto de artículos económicos y numerosas imitaciones de marcas de lujo [33].

En 2024, la app cambió de nombre para distanciarse de su imagen asociada a falsificaciones y evitar la presión legal de las marcas afectadas. Aun así, sigue centrada en precios reducidos y productos inspirados en diseños exclusivos. Hoy es una de las aplicaciones de compras más descargadas en España [34].

Datos recopilados

Amplio conjunto de datos:

- Información de identificación y contacto (nombre, correo, dirección, teléfono, identificador de usuario).
- Ubicación aproximada y precisa.
- Contenido generado por el usuario (fotos, vídeos).
- Historial de compras y métodos de pago.
- Información técnica (logs, diagnósticos).
- Interacciones y otras acciones en la app.

Propósitos: Se emplean de forma combinada para:

- Funcionamiento de la app.
- Personalización de la experiencia.
- Administración de cuentas.
- Análisis de comportamiento.
- Publicidad y marketing.

Medidas de seguridad: Admite la eliminación de datos y aplica cifrado en tránsito.

Aplicación YouTube Kids (com.google.android.apps.youtube.kids)

Aplicación diseñada específicamente para niños de hasta 12 años, que ofrece una experiencia segura y adaptada para que los más pequeños puedan explorar contenido audiovisual apropiado para su edad, donde los padres pueden gestionar el tiempo de pantalla, limitar el contenido que sus hijos pueden ver, bloquear videos y canales, y aprobar manualmente qué contenido está disponible [35].

Datos recopilados

Recoge:

- Grabaciones de voz.
- Ubicación aproximada.
- Identificadores y datos de contacto.
- Historial de búsqueda, interacciones y otras acciones.
- Información técnica (logs, diagnósticos).

Propósitos: Los datos se utilizan para el funcionamiento de la app, la personalización y el cumplimiento de políticas de seguridad infantil.

Medidas de seguridad: Incluye cifrado en tránsito, cumplimiento de la política de familias de Google Play y revisión por el programa MAVS.

Aplicación Vlad Nikita (me.apptivise.vladnikita)

La app está dirigida a niños en edad preescolar y primaria, buscando combinar entretenimiento con aprendizaje a través de rompecabezas, juegos de memoria, clasificación de formas y colores, y retos que estimulan la creatividad [36].

Datos recopilados

Recoge:

- Ubicación aproximada.
- Identificadores de dispositivo y usuario.
- Historial de compras.
- Interacciones, logs y otros datos de rendimiento.

Propósitos: Se utilizan con fines de análisis, personalización, seguridad y marketing.

Medidas de seguridad: No permite la eliminación de datos y no implementa cifrado en tránsito.

Aplicación Twitch (tv.twitch.android.app)

Plataforma de *streaming* en vivo, principalmente conocida por la retransmisión de videojuegos, aunque actualmente ofrece una gran variedad de contenidos como música, charlas, arte y eventos [37].

Datos recopilados

Incluye:

- Ubicación aproximada.
- Identificadores de usuario y dispositivo.
- Historial de búsqueda, interacciones, mensajes y contenido generado por el usuario.
- Información técnica (logs, diagnósticos).

Propósitos: Los datos se emplean para:

- Personalización de la experiencia.
- Análisis de uso.
- Publicidad dirigida.
- Cumplimiento normativo y seguridad.

Medidas de seguridad: Admite la eliminación de datos y aplica cifrado en tránsito.

5.5 Proceso de implementación

El proceso de implementación de la ontología DataSecOnt se llevó a cabo en un entorno controlado que garantizase tanto la persistencia de los datos como la capacidad de ejecutar consultas SPARQL y razonamiento semántico. Para ello, se realizaron una serie de fases que incluyeron la preparación del entorno técnico, el desarrollo del modelo ontológico y su posterior carga en el sistema de almacenamiento semántico GraphDB.

En primer lugar, se habilitó una máquina virtual basada en Ubuntu 22.04, configurada específicamente para disponer de un entorno gráfico ligero. Esta configuración permitió utilizar aplicaciones de escritorio, como navegadores web, necesarias para acceder de forma visual a la interfaz de administración de GraphDB. A fin de facilitar el acceso remoto, se configuró Google Remote

Desktop, lo que permitió trabajar de manera cómoda desde cualquier dispositivo con conexión a Internet [16].

Posteriormente, se procedió a crear un repositorio en GraphDB con perfil de razonamiento OWL2 RL, seleccionado por su equilibrio entre capacidad inferencial y eficiencia. Este repositorio fue el destino final de la carga de la ontología [15].

El desarrollo del archivo OWL que contiene la definición formal de DataSecOnt se realizó manualmente utilizando Visual Studio Code, siguiendo las directrices de la metodología *Ontology Development 101* [6]. Este enfoque permitió definir con precisión las clases, propiedades, restricciones y anotaciones que estructuran el modelo semántico.

Una vez completada y validada la creación del archivo OWL, este se importó en GraphDB a través del módulo de carga de datos. Tras la importación, se comprobó que el razonador se activaba correctamente y que el modelo era consistente, validando mediante consultas SPARQL que los axiomas definidos y las inferencias se producían según lo previsto. Este paso resultó fundamental para establecer la siguiente iteración del proceso, que consistió en refinar el archivo OWL, volver a importar la versión actualizada y verificar nuevamente su correcto funcionamiento.

Durante este proceso de implementación, la etapa en la que más dificultades he tenido ha sido durante las validaciones iniciales, la primera versión de la ontología era muy primitiva y aunque se podían hacer inferencias sobre ella, ha sido difícil "descubrir" que es lo que faltaba para completarla, aunque en la documentación de las versiones (ver sección 6.1.1 Posible refinamiento) no se reflejan demasiados cambios, se debe a que fue difícil identificarlos en un principio.

Todo este proceso, incluyendo los pasos detallados de instalación, configuración, desarrollo, carga y validación, así como las capturas de pantalla que ilustran cada etapa, se encuentra documentado de forma completa en el Anexo D – Proceso de implementación, al que se remite al lector para una consulta exhaustiva.

Capítulo 6 - Validación y evaluación

La validación y evaluación de la ontología DataSecOnt se llevó a cabo combinando diferentes estrategias complementarias que permitieron garantizar la coherencia lógica del modelo y su capacidad de respuesta frente a los objetivos planteados.

Por un lado, se realizó una inspección visual exhaustiva de los grafos generados en GraphDB tras la importación de cada versión**¡Error! No se encuentra el origen de la referencia.**. Esta revisión permitió identificar y corregir errores estructurales, como duplicidades de clases, relaciones redundantes o ausencias de anotaciones descriptivas que dificultaban la interpretación del modelo.

Por otro lado, se utilizaron de forma sistemática las CQs definidas en las fases iniciales del diseño (consultar 3.2.6.2 Requisitos funcionales: Preguntas de Competencia (*Competency Questions*)). Estas consultas SPARQL actuaron como principal mecanismo de verificación funcional, puesto que permitieron comprobar que la ontología respondía correctamente a los supuestos de uso y preguntas de competencia establecidos. La ejecución de estas consultas fue determinante para detectar inconsistencias semánticas, validar las inferencias automáticas generadas por el razonador y asegurar que la estructura conceptual reflejaba adecuadamente el dominio modelado.

Este proceso de revisión y contraste, basado en la alternancia de inspección visual y evaluación mediante *Competency Questions*, resultó clave en la iteración progresiva de versiones. Cada ciclo de refinamiento incorporó mejoras en la claridad, la granularidad y la alineación terminológica de la ontología, garantizando así su solidez y su aplicabilidad práctica.

6.1.1 Posible refinamiento

Como parte del proceso iterativo de mejora y validación, se realizaron diversos ciclos de refinamiento de la ontología, cada uno de los cuales permitió incrementar su claridad conceptual y su capacidad para responder de forma precisa a las *Competency Questions* definidas. A continuación, se describen las principales modificaciones realizadas en cada transición de versión, junto con sus motivaciones y los criterios empleados para su incorporación.

- **De la primera versión a la segunda**

En la transición de la primera a la segunda versión de la ontología se llevaron a cabo varias mejoras estructurales y de claridad. En primer lugar, se eliminó la clase Dato como clase padre de TipoDeDato, de manera que Dato pasó a ser una clase independiente, simplificando así la jerarquía. Asimismo, se eliminaron declaraciones duplicadas de la clase TipoDeDato, que aparecían declaradas dos veces de forma redundante. Se añadieron etiquetas (*labels*) y comentarios descriptivos a las clases y propiedades para mejorar la comprensión de la ontología. También se introdujeron nuevas propiedades y se reorganizaron los niveles de clasificación de los propósitos, permitiendo una representación más detallada de los fines de uso de los datos. Por último, se actualizaron relaciones y se eliminaron redundancias presentes en la versión inicial.

- **De la segunda versión a la tercera**

Para esta segunda versión, se utilizó como base un JSON con datos extraído mediante web scraping de la sección de Seguridad de los Datos de Google Play, con un total de 2122 aplicaciones. Este análisis permitió refinar y adaptar el vocabulario de la ontología a las categorías y prácticas reales que utilizan los desarrolladores. Por ejemplo, dentro del bloque de datos de salud, la clase que inicialmente se denominaba Información Estado Físico fue renombrada como InformacionFisica para alinearla con la terminología oficial de Google Play. En cuanto a las prácticas de seguridad, se detectó que faltaban

opciones relevantes como "*Data isn't encrypted"* y "*Data can't be deleted*", por lo que se crearon las clases NoCifrado y NoPermiteEliminacion.

Además, se identificó la necesidad de incorporar una característica esencial: la diferenciación explícita de si un dato se recoge o se comparte, mediante la propiedad modoUsoDato, que únicamente puede tomar los valores "recopilacion" o "comparticion". También se incorporaron otros atributos importantes como datoOpcional, procesamientoEfimero y datoEsSensitivo, que permiten caracterizar con mayor precisión cada tipo de dato. Estas propiedades se describieron en la documentación de la ontología para facilitar su interpretación y uso. Se adaptaron los ejemplos de instancias, estableciendo que las relaciones de propósito utilicen esRequeridoPara cuando se trate de recopilación y seCompartePara cuando se trate de compartición.

Durante las pruebas y consultas relacionadas con las *Competency Questions* (CQs), se detectaron casos especiales, como el de CorreoElectronico, que pertenece tanto a la categoría de datos personales como a la de mensajes. Para reflejar esta dualidad, se ajustó su definición de modo que heredara de ambas clases en lugar de crear duplicados conceptuales. Por otro lado, aunque la propiedad datoEsSensitivo ya existía, se estableció de forma explícita qué tipos de datos debían considerarse sensibles, siguiendo las indicaciones de Google [24]. Es importante destacar que estas propiedades de datos no se heredan automáticamente en OWL, por lo que fue necesario marcarlas de forma individual en cada subclase correspondiente.

- **De la tercera versión a la cuarta**

En esta etapa, se incorporaron nuevos ejemplos de instancias para ampliar la cobertura y mejorar la representatividad de la ontología. Estas nuevas instancias permitieron validar la robustez de la estructura diseñada y comprobar que las consultas SPARQL daban como resultado una representación fiel y detallada de los diferentes tipos de datos, sus propósitos de uso, el carácter sensible u opcional de la información, y las medidas de seguridad aplicadas.

- **De la cuarta versión a la quinta**

Finalmente, la quinta versión y la última incluye una modificación simple pero que impacta mucho en el modelo, durante todas las versiones anteriores se habían declarado como clases las instancias de ejemplo, así como sus datos recogidos, en esta versión final las instancias han sido declaradas como *NamedIndividual*, haciendo que se comporten como ejemplos y no como clases.

El enlace al repositorio que contiene todas las versiones de la ontología, se recoge para su consulta en el Anexo C – Código del TFG.

6.2 Consultas SPARQL

Para comprobar la capacidad de la ontología DataSecOnt de representar de forma adecuada el dominio y satisfacer los objetivos definidos, se diseñó y ejecutó un conjunto de consultas SPARQL. Estas consultas incluyen, por un lado, las preguntas de competencia (*Competency Questions*) establecidas durante la fase de especificación de requisitos, que permiten verificar que el modelo responde correctamente a los escenarios de uso previstos [17, 19].

Por otro lado, se desarrollaron consultas adicionales que, sin formar parte explícita de las preguntas de competencia, resultan igualmente relevantes para explorar, validar y ejemplificar la estructura semántica de la ontología. Este conjunto de consultas ha servido como base para la evaluación funcional y como mecanismo de verificación progresiva a lo largo de los distintos ciclos de refinamiento.

6.2.1 Consultas de competencia

1. **CQ1:** ¿Qué tipos de datos pueden recopilar las aplicaciones según la sección de Seguridad de los Datos?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?tipo ?label
WHERE {
 ?tipo rdfs:subClassOf+ ds:TipoDeDato .
 OPTIONAL { ?tipo rdfs:label ?label . }
}
ORDER BY ?tipo
```

Esta consulta tiene como objetivo identificar todos los tipos de datos definidos en la ontología que pueden ser objeto de recogida por parte de las aplicaciones, conforme a la estructura de categorías utilizada en la sección de Seguridad de los Datos de Google Play.

Para ello, se utilizan las relaciones de jerarquía (rdfs:subClassOf+) que permiten recorrer la taxonomía completa de tipos de datos hasta el nivel más específico. Así, se obtienen tanto las clases generales (por ejemplo, DatoPersonal) como las subclases concretas (por ejemplo, Telefono, Direccion, CorreoElectronico).

Se emplea SELECT DISTINCT para eliminar posibles duplicados derivados de la herencia múltiple.

2. **CQ2:** ¿Cómo se clasifican los datos recopilados por las aplicaciones?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?categoria ?categoriaLabel ?subtipo ?subtipoLabel
WHERE {
 ?categoria rdfs:subClassOf ds:TipoDeDato .
 OPTIONAL { ?categoria rdfs:label ?categoriaLabel . }

 OPTIONAL {
  ?subtipo rdfs:subClassOf ?categoria .
  FILTER(?subtipo != owl:Nothing)

  ?dato a ?subtipo .
  ?app ds:recopilaDato ?dato .

  OPTIONAL { ?subtipo rdfs:label ?subtipoLabel . }
 }
}
ORDER BY ?categoria ?subtipo
```

La segunda consulta permite identificar, para cada categoría principal de datos definida en la ontología (ds:TipoDeDato), los subtipos concretos de datos que efectivamente se recopilan en las aplicaciones modeladas en el grafo RDF.

El objetivo principal es mostrar la jerarquía práctica de la clasificación de datos recopilados por aplicaciones y de esta forma ver qué tipos de datos aparecen con instancias reales en las aplicaciones de ejemplo, frente a otros que podrían estar definidos pero no utilizados en los ejemplos.

La condición ?dato a ?subtipo y ?app ds:recopilaDato ?dato es la parte clave de la consulta y asegura que únicamente se muestren los subtipos que tienen instancias concretas de datos recogidos por al menos una aplicación. Si un subtipo existe en la ontología pero no hay ejemplos de su uso, no aparecerá en el resultado.

3. **CQ3:** ¿Cómo se clasifican los datos compartidos por las aplicaciones?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?categoria ?categoriaLabel ?subtipo ?subtipoLabel
WHERE {
 ?categoria rdfs:subClassOf ds:TipoDeDato .
 OPTIONAL { ?categoria rdfs:label ?categoriaLabel . }

 OPTIONAL {
  ?subtipo rdfs:subClassOf ?categoria .
  FILTER(?subtipo != owl:Nothing)

  ?dato a ?subtipo .
  ?dato ds:seCompartePara ?proposito .

  OPTIONAL { ?subtipo rdfs:label ?subtipoLabel . }
 }
}
ORDER BY ?categoria ?subtipo
```

El propósito de esta tercera consulta es imitar el comportamiento de la anterior pero para los subtipos concretos de datos que son compartidos por las aplicaciones.

Cabe destacar, que la diferencia con la consulta de recopilación es que aquí se comprueba explícitamente que existe un propósito de compartición (ds:seCompartePara), por lo que la consulta no devuelve datos meramente recogidos si no se comparten.

4. **CQ4:** ¿Qué medidas de seguridad declaran implementar las aplicaciones?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app ?appLabel ?permiteEliminacion ?seguridad ?seguridadLabel
WHERE {
 ?app a ds:Aplicacion .

 # Medida de seguridad
 ?app ds:implementaMedidaSeguridad ?seguridad .

 # Permite eliminación de datos
 ?app ds:permiteEliminacionDatos ?permiteEliminacion .

 OPTIONAL { ?app rdfs:label ?appLabel . }
 OPTIONAL { ?seguridad rdfs:label ?seguridadLabel . }
}
ORDER BY ?app ?seguridad
```

La cuarta consulta permite identificar las medidas de seguridad que cada aplicación declara implementar junto con la información sobre si permiten o no la eliminación de datos por parte del usuario.

El objetivo principal es visualizar de manera conjunta las prácticas de seguridad y la política de eliminación de datos de cada aplicación, proporcionando una visión más completa del nivel de compromiso con la privacidad y la protección de la información.

El triple ?app ds:permiteEliminacionDatos ?permiteEliminacion añade el dato booleano que especifica si la aplicación permite que los usuarios eliminen los datos que ha recopilado.

5. **CQ5:** ¿Qué aplicaciones recopilan datos personales sensibles?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app ?appLabel ?dato ?datoLabel ?tipo ?tipoLabel
WHERE {
 # Encontrar instancias de datos
 ?dato a ?tipo .

 # El dato tiene marcado que es sensitivo
 ?tipo ds:datoEsSensitivo "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .

 # La aplicación recopila el dato
 ?app ds:recopilaDato ?dato .

 # Opcional: etiqueta del dato
 OPTIONAL { ?dato rdfs:label ?datoLabel . }

 # Opcional: etiqueta de la clase del dato
 OPTIONAL { ?tipo rdfs:label ?tipoLabel . }

 # Opcional: etiqueta de la aplicación
 OPTIONAL { ?app rdfs:label ?appLabel . }
}
ORDER BY ?app ?tipo
```

La quinta consulta permite identificar qué aplicaciones recopilan datos personales que han sido marcados como sensibles en la ontología, mostrando tanto la clase del dato como la instancia concreta que se recoge.

El objetivo principal es obtener un listado de aplicaciones que manejan información con mayor nivel de riesgo para la privacidad, como identificadores, datos de ubicación precisa o información financiera.

El triple ?tipo ds:datoEsSensitivo "true" filtra únicamente aquellos tipos de datos que están clasificados como sensibles según la definición establecida en la ontología.

6. **CQ6:** ¿Qué aplicaciones recopilan datos personales y de qué tipo?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app ?appLabel ?dato ?datoLabel ?tipoDato ?tipoLabel
WHERE {
 # Cada dato recopilado por la app
```

```
    ?app ds:recopilaDato ?dato .

    # El dato es de tipo personal (subclase de DatoPersonal)
    ?dato a ?tipoDato .
    ?tipoDato rdfs:subClassOf+ ds:DatoPersonal .

    # Etiqueta del tipo de dato (p. ej., Nombre, Dirección)
    OPTIONAL { ?tipoDato rdfs:label ?tipoLabel . }

    # Etiqueta del dato concreto
    OPTIONAL { ?dato rdfs:label ?datoLabel . }

    # Etiqueta de la aplicación
    OPTIONAL { ?app rdfs:label ?appLabel . }
  }
  ORDER BY ?app ?tipoDato
```

La sexta consulta permite identificar qué aplicaciones recopilan datos personales y de qué tipo, mostrando tanto la categoría semántica (por ejemplo, Nombre, Dirección) como la instancia concreta de cada dato recogido.

El objetivo es ofrecer una visión detallada de las aplicaciones que manejan información personal y conocer exactamente cuáles son las clases de datos implicadas. El patrón ?tipoDato rdfs:subClassOf+ ds:DatoPersonal garantiza que solo se consideren aquellos datos que pertenecen a la jerarquía de datos personales definida en la ontología.

7. **CQ7:** ¿De qué formas se comparten los datos de las aplicaciones?

```
  PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

  SELECT
DISTINCT ?app ?appLabel ?dato ?datoLabel ?proposito ?propositoLabel ?modoUso
  WHERE {
    # Datos que son recopilados o compartidos por alguna aplicación
    ?app ds:recopilaDato ?dato .

    # Solamente los que se comparten
    ?dato ds:modoUsoDato "comparticion" .

    OPTIONAL { ?dato ds:seCompartePara ?proposito . }
    OPTIONAL { ?dato ds:modoUsoDato ?modoUso . }
    OPTIONAL { ?app rdfs:label ?appLabel . }
    OPTIONAL { ?dato rdfs:label ?datoLabel . }
    OPTIONAL { ?proposito rdfs:label ?propositoLabel . }
  }
  ORDER BY ?app ?dato
```

Esta consulta permite identificar de qué formas se comparten los datos de las aplicaciones, mostrando para cada dato compartido el propósito asociado (por ejemplo, publicidad o cumplimiento legal) y el modo de uso declarado.

La parte principal de esta consulta es la condición ?dato ds:modoUsoDato "comparticion" restringe los resultados exclusivamente a los datos que efectivamente se comparten con terceros.

8. **CQ8:** ¿Cuáles son los distintos fines por los que se recopilan datos?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?proposito ?label
WHERE {
 ?dato ds:esRequeridoPara ?proposito .
 OPTIONAL { ?proposito rdfs:label ?label . }
}
ORDER BY ?proposito
```

La octava consulta tiene como finalidad recopilar el conjunto de propósitos declarados que justifican la recogida de datos por parte de las aplicaciones. En este caso, se extraen todos los valores distintos de ds:esRequeridoPara, que vinculan cada dato con su finalidad (por ejemplo, funcionalidad de la aplicación, personalización o cumplimiento legal).

La consulta ofrece así un listado completo de los motivos por los que se recogen datos, permitiendo analizar y clasificar estos fines de manera global.

9. **CQ9:** ¿Cuáles son los distintos fines por los que se comparten datos?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?proposito ?label
WHERE {
 ?dato ds:seCompartePara ?proposito .
 OPTIONAL { ?proposito rdfs:label ?label . }
}
ORDER BY ?proposito
```

Esta consulta tiene el mismo comportamiento que la anterior pero enfocándose en los fines por los que se comparten datos, de la misma forma que la consulta anterior pero usando ds:seCompartePara, obteniendo un listado de los fines.

10. **CQ10:** ¿Qué aplicaciones permiten la eliminación de datos?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?app
WHERE {
 ?app a ds:Aplicacion .
 ?app                                                    ds:permiteEliminacionDatos
"true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
}
ORDER BY ?app
```

Esta consulta tiene como objetivo hacer un listado con el fin de analizar y clasificar aquellas aplicaciones que permitan la eliminación de datos, para ello se utiliza la cláusula ?app ds:permiteEliminacionDatos "true"^^http://www.w3.org/2001/XMLSchema#boolean, que indica que para que una aplicación permita eliminar los datos se tiene que dar que la propiedad ds:permiteEliminacionDatos está marcada como verdadera (*true*).

11. **CQ11:** ¿Qué aplicaciones recopilan datos que son opcionales?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app ?dato ?label
WHERE {
 ?app ds:recopilaDato ?dato .
 ?dato ds:datoOpcional "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
 OPTIONAL { ?dato rdfs:label ?label . }
}
ORDER BY ?app ?dato
```

La siguiente consulta se centra en listar las aplicaciones que recopilan datos opcionales, para ello se sigue la estructura de la anterior consulta pero usando la propiedad ds:datoOpcional, que tiene que estar marcada como verdadera (*true*).

12. **CQ12:** ¿Qué tipos de datos se consideran sensibles según la ontología?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?tipo ?label
WHERE {
 ?tipo ds:datoEsSensitivo "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
 OPTIONAL { ?tipo rdfs:label ?label . }
}
ORDER BY ?tipo
```

La consulta que se ha planteado permite identificar todos los tipos de datos que la ontología define explícitamente como sensibles. Para ello, se filtran aquellas clases de datos que tienen la propiedad ds:datoEsSensitivo marcada como verdadera, de acuerdo con los criterios de la sección de Seguridad de los Datos de Google Play. El resultado es un listado de las categorías de información que requieren especial atención por su carácter delicado, junto con sus etiquetas descriptivas si están disponibles.

13. **CQ13:** ¿Qué tipos de datos tienen múltiples propósitos de uso?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?dato (COUNT(DISTINCT ?proposito) AS ?numeroDePropositos)
WHERE {
 ?dato ds:esRequeridoPara ?proposito .
}
GROUP BY ?dato
HAVING(COUNT(DISTINCT ?proposito) > 1)
ORDER BY DESC(?numeroDePropositos)
```

La decimotercera consulta tiene como objetivo identificar qué datos están asociados a más de un propósito de uso dentro de las aplicaciones.

Esto permite detectar aquellos datos que cumplen funciones múltiples (por ejemplo, un mismo dato utilizado tanto para el funcionamiento de la aplicación como para fines analíticos o de personalización).

Se recuperan todos los datos que tienen alguna relación ds:esRequeridoPara con un propósito concreto (?proposito), se agrupan los resultados por el propósito se calculan cuantos propósitos

distintos se asocian a cada dato usando COUNT y finalmente se conservan en el ranking aquellas con más de un propósito.

14. **CQ14:** ¿Qué datos no se comparten con terceros?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?dato
WHERE {
 ?app ds:recopilaDato ?dato .
 FILTER NOT EXISTS { ?dato ds:seCompartePara ?proposito . }
}
ORDER BY ?dato
```

Esta consulta está diseñada para identificar los datos que se recopilan pero no se comparten con terceros. Funciona de la siguiente forma:

Localiza todos los datos (?dato) que aparecen como recopilados por alguna aplicación. Usa FILTER NOT EXISTS para excluir aquellos datos que tengan asociada una relación ds:seCompartePara, es decir, que indiquen algún propósito de compartición. El resultado es el listado de datos que permanecen en uso interno de la aplicación y que no se transfieren a otros destinatarios.

15. **CQ15:** ¿Qué aplicaciones utilizan cifrado en tránsito como medida de seguridad?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app
WHERE {
 ?app a ds:Aplicacion .
 ?app ds:implementaMedidaSeguridad ds:CifradoEnTransito .
}
ORDER BY ?app
```

Finalmente esta consulta permite obtener el listado de aplicaciones que declaran utilizar cifrado en tránsito como medida de seguridad. Para ello se recuperan todas las instancias de aplicaciones (?app) con ?app a ds:Aplicación y se filtran aquellas que tienen explícitamente asociada la relación ds:implementaMedidaSeguridad con la clase ds:CifradoEnTransito.

Después de haber realizado estas consultas SPARQL con la última versión de la ontología y tras haber analizado los resultados para los ejemplos de aplicaciones que se han incluido, concluimos la fase de refinamiento de la ontología y se establecerá la quinta versión como la versión final para este trabajo.

6.2.2 Otras consultas

Estas consultas complementarias se diseñaron con el objetivo de explorar distintas perspectivas sobre los datos modelados en la ontología, así como ilustrar su aplicabilidad a casos de uso reales. A diferencia de las preguntas de competencia, que responden a los requisitos definidos al inicio del proyecto, estas consultas adicionales permiten:

- Identificar patrones frecuentes de recopilación y uso de datos.
- Detectar características específicas (por ejemplo, datos opcionales o sensibles).
- Generar rankings y agregaciones que faciliten el análisis comparativo.

La ejecución de estas consultas contribuyó a validar la completitud de la ontología y a demostrar su capacidad de respuesta ante interrogantes prácticos que podrían surgir en un escenario de explotación real de la información como la realización de rankings o listados más exhaustivos.

1. ¿Qué aplicaciones comparten datos con fines de marketing?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app ?dato
WHERE {
 ?app ds:recopilaDato ?dato .
 ?dato ds:seCompartePara ds:PublicidadOMarketing .
}
ORDER BY ?app ?dato
```

2. ¿Qué datos se utilizan con fines de prevención de fraude, seguridad y cumplimiento?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?dato
WHERE {
 ?dato ds:esRequeridoPara ds:SeguridadCumplimientoYPrevencion .
}
ORDER BY ?dato
```

3. ¿Qué datos se utilizan para personalización?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?dato
WHERE {
 ?dato ds:esRequeridoPara ds:Personalizacion .
}
ORDER BY ?dato
```

4. ¿Qué datos tienen marcado que son opcionales y sensibles al mismo tiempo?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?dato ?label
WHERE {
 ?dato a ?tipo .
 ?tipo ds:datoEsSensitivo "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
 ?dato ds:datoOpcional "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
 OPTIONAL { ?dato rdfs:label ?label . }
}
ORDER BY ?dato
```

5. ¿Qué aplicaciones utilizan identificadores de dispositivo y con qué fines?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?app ?dato ?proposito
```

```
WHERE {
  ?app ds:recopilaDato ?dato .
  ?dato a ds:IDDispositivo .
  OPTIONAL { ?dato ds:esRequeridoPara ?proposito . }
}
ORDER BY ?app ?dato
```

6. ¿Qué aplicaciones recopilan datos de ubicación (precisa o aproximada)?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?app ?dato ?tipo
WHERE {
  ?app ds:recopilaDato ?dato .
  ?dato a ?tipo .
  ?tipo rdfs:subClassOf+ ds:DatoUbicacion .
}
ORDER BY ?app ?dato
```

7. ¿Qué datos opcionales se utilizan con fines de personalización?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?dato
WHERE {
  ?dato ds:esRequeridoPara ds:Personalizacion .
  ?dato ds:datoOpcional "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
}
ORDER BY ?dato
```

8. ¿Qué aplicaciones no permiten la eliminación de datos?

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>

SELECT DISTINCT ?app
WHERE {
  ?app a ds:Aplicacion .
  ?app                                                    ds:permiteEliminacionDatos
"false"^^<http://www.w3.org/2001/XMLSchema#boolean> .
}
ORDER BY ?app
```

9. Ranking de propósitos de recopilación más frecuentes

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?proposito ?propositoLabel (COUNT(?dato) AS ?numeroDeDatos)
WHERE {
  ?dato ds:esRequeridoPara ?proposito .
  OPTIONAL { ?proposito rdfs:label ?propositoLabel . }
}
GROUP BY ?proposito ?propositoLabel
ORDER BY DESC(?numeroDeDatos)
```

10. Ranking de propósitos de compartición más frecuentes

```
PREFIX ds: <http://tfg.uva.es/ontologias/datasecont#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?proposito ?propositoLabel (COUNT(?dato) AS ?numeroDeDatos)
WHERE {
  ?dato ds:seCompartePara ?proposito .
  OPTIONAL { ?proposito rdfs:label ?propositoLabel . }
}
GROUP BY ?proposito ?propositoLabel
ORDER BY DESC(?numeroDeDatos)
```

El conjunto de consultas anteriores refleja algunos ejemplos de explotación avanzada de la ontología DataSecOnt. Todas ellas se ejecutaron satisfactoriamente sobre la última versión del modelo.

Capítulo 7 - Resultados y análisis

7.1 Resultados obtenidos

La última versión de la ontología DataSecOnt se cargó y validó de forma completa en el repositorio de GraphDB, permitiendo la ejecución de consultas SPARQL y la exploración visual de su estructura.

A continuación, se presentan los principales resultados obtenidos, organizados en dos bloques complementarios:

- **Visualización de la estructura semántica**: se incluyen los gráficos generados por GraphDB, que muestran la jerarquía de clases, las relaciones definidas entre conceptos y la red de propiedades.
- **Análisis de los resultados de las consultas SPARQL**: se analizan de forma general los resultados obtenidos tanto con las *Competency Questions* como con las consultas adicionales diseñadas.

7.1.1 Visualización de la ontología

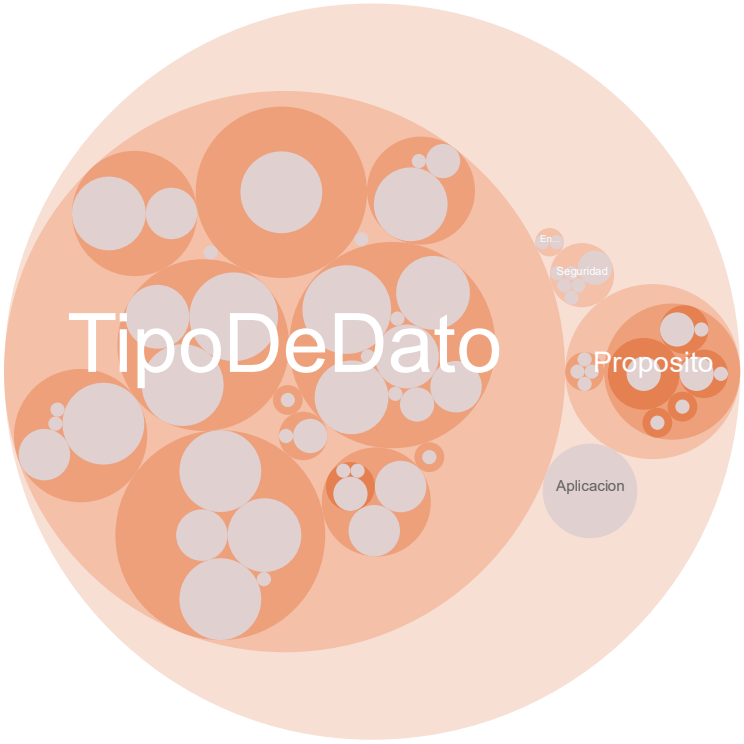

ILUSTRACIÓN 5: JERARQUÍA DE CLASES

La Ilustración 5: jerarquía de clases, muestra la jerarquía completa de clases de la ontología DataSecOnt, generada automáticamente en GraphDB. En esta representación, cada círculo corresponde a una clase del modelo, y su tamaño refleja el número relativo de subclases asociadas.

Se observa que el núcleo principal de la jerarquía está formado por la clase TipoDeDato, que agrupa la mayoría de las subclases y concentra un volumen significativo de instancias. A su alrededor se distribuyen otras clases principales como Proposito, Aplicacion y Seguridad, que estructuran las dimensiones fundamentales del modelo semántico: qué datos se recogen o comparten, con qué finalidad, por qué aplicaciones y con qué medidas de seguridad.

El hecho de que algunas clases aparezcan con menor tamaño o situadas en la periferia se debe a que contienen un número más reducido de elementos subordinados o instancias. Esta disposición visual

facilita identificar de un vistazo qué conceptos son más centrales y cuáles presentan una granularidad menor.

Cabe destacar que en esta representación, no se muestra cada instancia definida en la, este aspecto no indica una falta de conexión semántica entre las instancias y las clases, sino que este gráfico se centra únicamente en visualizar el comportamiento jerárquico entre clases, no instancias.

Por este motivo, la visualización debe interpretarse principalmente como un mapa de jerarquía de clases, mientras que la verificación de las instancias se debe hacer mediante consultas SPARQL.

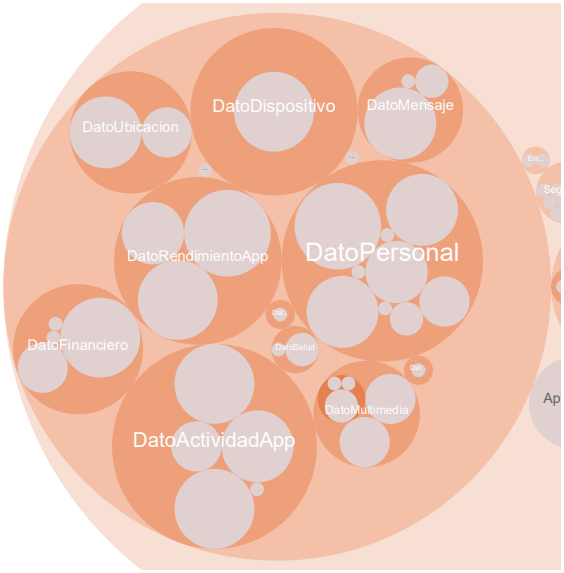

ILUSTRACIÓN 6: JERARQUÍA PARA TIPO DE DATO

La Ilustración 6: jerarquía para , muestra un detalle de la jerarquía de subclases de TipoDeDato, que constituye uno de los núcleos conceptuales más relevantes de la ontología DataSecOnt. En esta representación, cada círculo corresponde a una subclase de dato, y su tamaño refleja la cantidad relativa de instancias o subconceptos definidos dentro de cada categoría.

Se observa que la clase TipoDeDato se ramifica en diversas categorías que representan los principales tipos de información susceptibles de ser recogidos o compartidos por las aplicaciones, siendo DatoPersonal una de las más relevantes con múltiples subclases en su interior.

Este esquema visual facilita apreciar la distribución de granularidad dentro de la clase TipoDeDato y permite identificar qué categorías cuentan con una mayor riqueza de subclases. Al igual que en los gráficos anteriores, esta vista no representa las relaciones, sino únicamente la estructura taxonómica de la información.

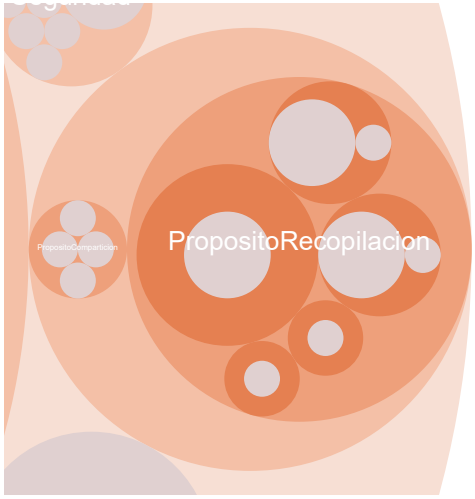

ILUSTRACIÓN 7: JERARQUÍA PARA PROPOSITO

La Ilustración 7: jerarquía para proposito**¡Error! No se encuentra el origen de la referencia.**, muestra la jerarquía de clases de propósito de uso definida en la ontología DataSecOnt. Este componente modela las razones principales por las que las aplicaciones recopilan o comparten datos de los usuarios.

Cada círculo interno corresponde a un subpropósito o uso concreto, por ejemplo, dentro de PropositoRecopilacion, se encuentra PropositoFuncional e internamente puede ser o FuncionesDeLaApp o bien ComunicacionesDelDesarrolldor, en esta jerarquía el tamaño de los círculos refleja la cantidad relativa de relaciones e instancias que hacen referencia a cada uno.

De la misma forma que para Proposito y las jerarquías que ya hemos comentado, en la Ilustración 8: jerarquía para seguridad, se muestra la jerarquía para Seguridad, una de las subclases más pequeñas que tiene 6 instancias en su interior relativas a las medidas de seguridad que toman los desarrolladores de las aplicaciones con los datos.

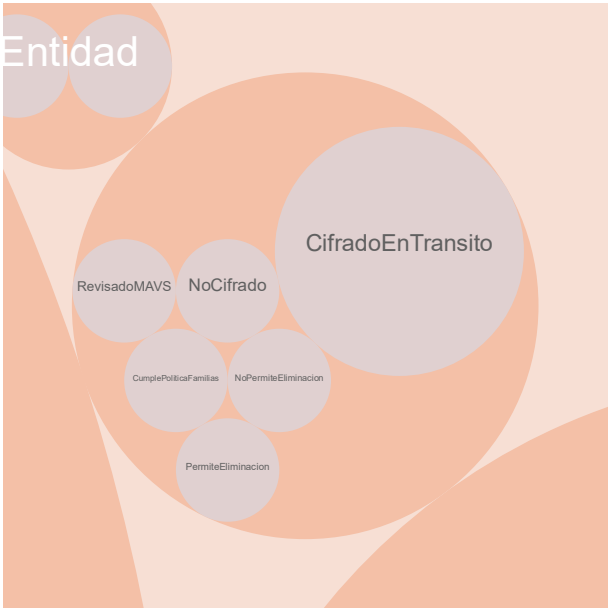

ILUSTRACIÓN 8: JERARQUÍA PARA SEGURIDAD

La Ilustración 9: gráfico de dependencias, muestra el diagrama de dependencias o *chord diagram* generado automáticamente en GraphDB [15]. Este tipo de visualización permite identificar de manera intuitiva las relaciones entre distintas entidades de la ontología, tanto clases como instancias.

Cada segmento del círculo representa un concepto o recurso, mientras que las cuerdas internas indican las relaciones existentes entre ellos. El grosor de cada cuerda refleja la cantidad de conexiones que vinculan dos elementos, de modo que las relaciones más frecuentes aparecen representadas con mayor anchura.

Este diagrama evidencia la interconexión existente entre los principales componentes del modelo, como los tipos de datos, los propósitos de uso, las aplicaciones y las prácticas de seguridad. En concreto, se observa que:

- Existen múltiples relaciones cruzadas entre datos personales y propósitos de recopilación y compartición.
- Determinados tipos de dato (por ejemplo, identificadores de dispositivo) mantienen enlaces con varias finalidades de uso.
- La red semántica es densa, lo que confirma que la ontología no está compuesta por segmentos aislados, sino que constituye un grafo coherente e integrado.

Varias cuerdas conectan un mismo segmento con múltiples destinos, lo que indica que una misma entidad se relaciona con diversos elementos. Por ejemplo: IDUsuario está relacionado con FuncionesDeLaApp, Analisis y PublicidadOMarketing.

Ningún segmento permanece completamente sin cuerdas. Esto es relevante, ya que valida que todas las clases principales definidas en el modelo participan en relaciones con otras entidades, lo que demuestra la coherencia de la ontología.

A diferencia de las representaciones jerárquicas, que únicamente ilustran la clasificación taxonómica de conceptos, el diagrama de dependencias refleja cómo se relacionan en la práctica las clases y las instancias a través de propiedades de objeto, como la asociación de un tipo de dato con un propósito específico o la vinculación de una aplicación con los datos que recopila.

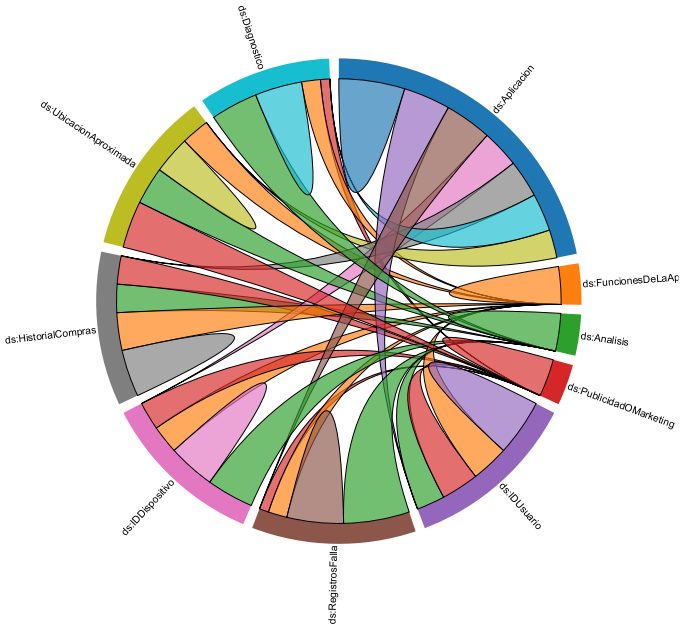

ILUSTRACIÓN 9: GRÁFICO DE DEPENDENCIAS

Los *visual graphs* generados en GraphDB para representar la red de conexiones de clases específicas [15], como TipoDeDato, Proposito, Aplicacion y otras, se incluyen en el Anexo E – Visual graphs, Estos diagramas muestran la red de relaciones de cada categoría principal de manera detallada y

resultan de interés complementario, aunque su comportamiento visual es similar al de los gráficos de jerarquías descritos en este capítulo.

7.1.2 Análisis de los resultados de las consultas SPARQL

Las consultas SPARQL ejecutadas sobre la ontología DataSecOnt fueron diseñadas para validar su capacidad de respuesta frente a los objetivos definidos y comprobar su operatividad en un entorno funcional. Estas consultas se agruparon en dos bloques principales:

- Por un lado, las *Competency Questions* (CQs), formuladas durante las fases iniciales del proyecto como criterios de validación funcional del modelo.
- Por otro, una serie de consultas complementarias orientadas a explorar en mayor profundidad la información representada por la ontología.

La ejecución de ambos grupos de consultas se realizó sobre la última versión cargada en GraphDB.

7.1.2.1 Consultas sobre las *Competency Questions*

El conjunto de quince CQs permitió validar de forma exhaustiva el comportamiento de la ontología y demostrar su capacidad de recuperar información coherente sobre los distintos aspectos de los datos gestionados por las aplicaciones. En términos generales, los resultados confirman que el modelo es consistente, completo y alineado con los objetivos de representación definidos.

Puede destacarse que:

- CQ1, CQ2 y CQ3 devolvieron correctamente la jerarquía completa de tipos de datos, tanto en términos de clasificación principal como de subtipos específicos asociados a las instancias de datos recopilados y compartidos. Estos resultados permitieron constatar que las relaciones taxonómicas (rdfs:subClassOf) se encuentran definidas de manera adecuada y que las etiquetas descriptivas se han cargado correctamente.
- CQ4 identificó de forma satisfactoria las medidas de seguridad declaradas por las aplicaciones, así como la propiedad que indica si permiten o no la eliminación de datos. Este resultado confirma que las propiedades permiteEliminacionDatos e implementaMedidaSeguridad están correctamente aplicadas a las instancias de Aplicacion.
- CQ5 y CQ6 proporcionaron listados de aplicaciones que recopilan datos personales y datos sensibles, confirmando que la propiedad datoEsSensitivo discrimina adecuadamente este atributo en las clases correspondientes.
- CQ7 permitió identificar las aplicaciones que efectivamente comparten datos con terceros, mostrando el valor comparticion en la propiedad modoUsoDato y los propósitos de compartición asociados.
- CQ8 y CQ9 devolvieron el conjunto de fines de recopilación y compartición contemplados en el modelo, evidenciando que la ontología recoge una variedad amplia de propósitos coherentes con las categorías identificadas en la sección de Seguridad de los Datos de Google Play.
- CQ10 y CQ15 permitieron extraer listados de aplicaciones con determinadas características de seguridad: por un lado, aquellas que permiten la eliminación de datos y, por otro, las que implementan cifrado en tránsito. Estos resultados validan la correcta aplicación de propiedades booleanas y relaciones con clases de medidas de seguridad.
- CQ11 y CQ12 identificaron, respectivamente, los datos opcionales y los tipos de datos considerados sensibles, mostrando que los atributos booleanos datoOpcional y, de nuevo, datoEsSensitivo están correctamente definidos y almacenados.
- CQ13 resultó especialmente relevante al mostrar datos que presentan múltiples propósitos de uso, confirmando que la ontología soporta relaciones n-arias y puede modelar casos en los que un mismo dato sirve a finalidades diferentes.

- CQ14 permitió verificar qué datos recopilados por las aplicaciones no se comparten con terceros, aspecto fundamental para discriminar los usos internos de los usos externos de la información.

Aunque la mayoría de las consultas confirmaron de manera directa el correcto funcionamiento de la ontología, conviene destacar dos aspectos especialmente significativos:

CQ13 (datos con múltiples propósitos de uso)

El resultado de esta consulta puso de manifiesto que un porcentaje significativo de datos se utilizan simultáneamente para varias finalidades, lo que valida la capacidad del modelo para representar escenarios de uso complejos en los que una misma categoría de información puede estar asociada a diferentes procesos, como la personalización y la publicidad. Este hallazgo es particularmente importante de cara a la explotación de la ontología en análisis de cumplimiento normativo o evaluación de riesgos.

CQ7 (formas de compartición de datos)

La consulta evidenció la correcta diferenciación entre recopilación y compartición mediante la propiedad modoUsoDato, mostrando que los datos marcados como comparticion están asociados a propósitos distintos de los de uso interno. Este resultado valida un componente central del modelo, orientado a reflejar la trazabilidad de los datos en todo su ciclo de tratamiento.

7.1.2.2 Consultas extra

Se ejecutó un conjunto de consultas complementarias orientadas a explorar en mayor detalle la capacidad de la ontología para recuperar información específica sobre patrones de uso, medidas de seguridad y finalidades de tratamiento de los datos.

Las consultas realizadas se encuentran en la sección 6.2.2 Otras consultas.

A continuación, se destacan los aspectos más relevantes:

En la consulta 1, se identificó que únicamente tres aplicaciones de todas las instancias incluidas en la ontología recopilan datos que posteriormente son compartidos con fines de marketing: Bonpreu, Pinterest y SaraMart.

La consulta número 6, devolvió los datos de ubicación recopilados por las aplicaciones, discriminando correctamente entre tipos de localización (precisa o aproximada). Entre los ejemplos definidos en la ontología, únicamente las aplicaciones Pinterest y SaraMart aparecen recogiendo la ubicación precisa.

En la séptima consulta, que pretendía identificar datos opcionales utilizados con fines de personalización, no devolvió ningún resultado. Este hecho confirma que en la versión actual de la ontología no existen instancias que cumplan simultáneamente ambos criterios, validando que la consulta discrimina correctamente en ausencia de coincidencias.

Finalmente, el hallazgo más destacable corresponde a la consulta 23, en la que se analizó qué aplicaciones no permiten la eliminación de datos. La única aplicación que cumple este criterio es Vlad y Nikita, una app dirigida al público infantil que, pese a contar con la certificación de familias [36], recopila información como ubicación aproximada, identificadores de dispositivo y usuario, historial de compras, interacciones, logs y otros datos de rendimiento. Estos datos se utilizan con fines de análisis, personalización y marketing, y no se aplica cifrado en tránsito. El hecho de que una aplicación orientada a niños recoja tanta información, no implemente medidas básicas de seguridad y aun así obtenga la certificación MASVS resulta especialmente llamativo desde el punto de vista de la protección de datos y la responsabilidad ética en el desarrollo de aplicaciones.

Capítulo 8 - Conclusiones y líneas de trabajo futuras

8.1 Conclusiones

8.1.1. Conclusiones de desarrollo

Desde el punto de vista técnico, el trabajo ha alcanzado de manera satisfactoria los objetivos planteados inicialmente. El principal objetivo era diseñar, construir y validar una ontología capaz de representar de forma detallada la sección de Seguridad de los Datos de Google Play.

A lo largo del proceso se ha implementado un modelo RDF/OWL completo que integra diversas clases, propiedades y restricciones, junto con un conjunto de instancias reales obtenidas mediante técnicas de web scraping aplicadas a Google Play. Este modelo se cargó en un repositorio funcional de GraphDB, y a partir de dicho repositorio, mediante la exploración de los gráficos generados y la ejecución de consultas SPARQL, se procedió al refinamiento progresivo de la ontología para mejorar su coherencia.

Entre los retos técnicos más relevantes resueltos destacan la extracción y carga de información procedente de fuentes externas como Google Play para poder obtener todo el vocabulario y relaciones relevantes para la ontología, la diferenciación explícita entre la recopilación y la compartición de datos, la definición de atributos booleanos (por ejemplo, datoEsSensitivo, datoOpcional) y su uso combinado en las consultas SPARQL, y la correcta clasificación jerárquica de los tipos de dato.

Asimismo, se desarrolló un conjunto exhaustivo de consultas SPARQL que han permitido verificar la consistencia del grafo, validar las relaciones definidas y demostrar la capacidad del modelo para responder a las *Competency Questions* y a otras consultas de interés.

Como principales limitaciones, se ha identificado la dificultad de representar de forma visual y comprensible un grafo tan extenso mediante las herramientas estándar de GraphDB, la necesidad de actualizar manualmente las instancias cuando se introducen cambios en la taxonomía, y la ausencia de un mecanismo de sincronización automática con nuevas fuentes de datos. No obstante, estas limitaciones no han impedido alcanzar los objetivos de desarrollo previstos ni comprobar la validez del enfoque propuesto.

8.1.2 Conclusiones académicas

Desde el punto de vista académico, el desarrollo de este proyecto ha supuesto un aprendizaje profundo en el ámbito de la representación del conocimiento y la Web Semántica, así como en la aplicación de estos enfoques a la modelización de dominios complejos como la protección de datos.

A nivel conceptual, el trabajo ha permitido comprender en detalle la utilidad de los lenguajes RDF y OWL para describir estructuras de información, así como la importancia de diseñar jerarquías de clases, propiedades y restricciones que garanticen la coherencia semántica del modelo. Además, se ha adquirido experiencia en el uso de las consultas SPARQL como herramienta para validar ontologías y obtener conocimiento a partir de los datos representados.

Este trabajo aporta un ejemplo práctico de cómo los principios de la Web Semántica pueden aplicarse en contextos de transparencia informativa y cumplimiento normativo, demostrando que es posible estructurar aspectos que tradicionalmente se presentan de manera textual y dispersa, como las políticas de privacidad.

el proyecto ha permitido adquirir un conocimiento sólido sobre el uso de GraphDB como plataforma de almacenamiento y gestión de grafos RDF. Durante el proceso se han utilizado sus funcionalidades principales, incluyendo la configuración de repositorios con diferentes perfiles de razonamiento, la

carga de datos en formatos RDF/XML, la ejecución de consultas SPARQL y la interpretación de los resultados tanto en forma de tablas como mediante visualizaciones gráficas. Este aprendizaje ha puesto de manifiesto tanto el potencial de esta herramienta para soportar modelos semánticos complejos como algunas de sus limitaciones prácticas.

8.2 Trabajo futuro

A partir de la experiencia adquirida durante el desarrollo de este proyecto, se identifican diversas líneas de mejora y posibles extensiones que podrían enriquecer y ampliar el alcance de la ontología DataSecOnt.

En primer lugar, sería recomendable trabajar en la ampliación del conjunto de instancias y casos de uso, incorporando datos procedentes de un número mayor de aplicaciones reales. De este modo, se podría validar con mayor amplitud la robustez de la ontología y garantizar que cubre un espectro representativo de prácticas de recogida, compartición y protección de datos.

En cuanto a funcionalidades avanzadas, una línea de desarrollo especialmente relevante sería la creación de un agente inteligente capaz de interpretar consultas en lenguaje natural. Este agente actuaría como interfaz conversacional que traduzca preguntas expresadas en lenguaje común a consultas SPARQL, de forma que usuarios sin conocimientos técnicos puedan consultar la información de la ontología de manera accesible.

Por último, otra vía de evolución interesante sería la integración con otras ontologías complementarias, como la desarrollada por el grupo App-PI, que modela los permisos declarados por las aplicaciones en Google Play. La combinación de ambos modelos permitiría construir un ecosistema de conocimiento más completo, capaz de correlacionar los permisos solicitados con los tipos de datos efectivamente recopilados y las finalidades declaradas, aportando un enfoque más detallado al análisis de riesgos y cumplimiento normativo.

9. Bibliografía

[1] Google. (2025). Ayuda de Play Console. Obtenido de Proporciona información para la sección de Seguridad de los datos de Google Play: https://support.google.com/googleplay/android-developer/answer/10787469?hl=es-419

[2] European Union. (2016). General Data Protection Regulation (GDPR). Official Journal of the European Union, L119, 1–88.

[3] Kohne, J. (2015). Ontology, its origins and its meaning in information science. En Philosophy, computing and information science (págs. 85-89). Routledge.

[4] Grado en Ingeniería Informática, U. d. (2024). Ingeniería del Conocimiento.

[5] Ontotext. (23 de Enero de 2025). GraphDB documentation. Obtenido de Ontologies: https://graphdb.ontotext.com/documentation/10.3/ontologies.html

[6] Noy, N. F., & Guinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University.

[7] Google. (2025). Ayuda de Google Play. Obtenido de Comprende las prácticas de privacidad y seguridad de las apps con la sección de Seguridad de los datos de Google Play: https://support.google.com/googleplay/answer/11416267?hl=es-419&co=GENIE.Platform%3DAndroid

[8] Filtran línea datos de 533 millones de usuarios de Facebook. (4 de Abril de 2021). DW. Obtenido de https://www.dw.com/es/filtran-datos-de-533-millones-de-usuarios-de-facebook-en-foro-de-ciberpiratas/a-57095450

[9] Higuera, A. (2 de Junio de 2023). Prueban que TikTok almacena los datos de todos sus usuarios en China. 20 minutos. Obtenido de https://www.20minutos.es/tecnologia/aplicaciones/tiktok-almacena-datos-todos-usuarios-china-5133843/

[10] PortalTIC. (30 de Enero de 2025). Google Play evita la publicación de 2,36 millones de apps que violaban las políticas de la plataforma. Europa Press. Obtenido de https://www.europapress.es/portaltic/ciberseguridad/noticia-google-play-evita-publicacion-236-millones-apps-violaban-politicas-plataforma-20250130121523.html

[11] Pandit, H. J., Esteves, B., Krog, G. P., Ryan, P., Golpayegani, D., & Flake, J. (2024). Data Privacy Vocabulary (DPV)--Version 2. arXiv preprint arXiv:2404.13426.

[12] Sacco, O., & Passant, A. (2011). A Privacy Preference Ontology (PPO) for Linked Data. Digital Enterprise Research Institute, National University of Ireland, Galway, 813.

[13] App-PI. (2024). Appi-PI Ontologies & SKOS vocabularies for Android applications.

[14] Grado en Ingeniería Informática, U. d. (2024). Sistemas Avanzados de Integración de Información.

[15] Ontotext. (2023). GraphDB Documentation Release 10.3.3. Obtenido de https://graphdb.ontotext.com/documentation/10.3/pdf/GraphDB.pdf

[16] Ubuntu. (2025). Ubuntu on GCP documentation. Obtenido de Launch an Ubuntu desktop on a VM: https://documentation.ubuntu.com/gcp/google-how-to/gce/launch-ubuntu-desktop/

[17] Ontotext. (23 de Enero de 2025). GraphDB documentation. Obtenido de SPARQL: https://graphdb.ontotext.com/documentation/10.3/sparql.html

[18] Draw.io. (2025). draw.io Documentation. Obtenido de: https://www.drawio.com/doc/

[19] Suárez-Figueroa, M. C., Gómez Pérez, A., & Villazón-Terrazas, B. (2009). How to Write and Use the Ontology Requirements Specification Document. Ontology Engineering Group, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid.

[20] Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. The knowledge engineering review, 11(2), 93-136.

[21] Grado en Ingeniería Informática, Universidad de Valladolid. (2025). Planificación y Desarrollo de Sistemas Computacionales.

[22] Hughes, B. (2009). Software project management (Quinta ed.). (M. Cotterell, Ed.) McGraw-Hill.

[23] Grado en Ingeniería Informática, Universidad de Valladolid. (2025). Profesión y Sociedad.

[24] Google. (2025). Ayuda de Play Console. Obtenido de Datos del Usuario: https://support.google.com/googleplay/android-developer/answer/10144311?hl=es-419&ref_topic=9877467&sjid=14628824234884710 31-EU#zippy=

[25] OWASP. (2024). OWASP MASVS (Mobile Application Security Verification Standard). OWASP. Obtenido de https://mas.owasp.org/MASVS/

[26] Google. (2025). Ayuda de Google Play. Obtenido de Usar Google Play Protect para proteger tus aplicaciones y la privacidad de tus datos: https://support.google.com/googleplay/answer/2812853?hl=es

[27] W3C. (2012). Best Practice Recipes for Publishing RDF Vocabularies. W3C Working Group Note. Obtenido de https://www.w3.org/TR/swbp-vocab-pub/

[28] W3C. (2014). RDF 1.1 XML Syntax. W3C Recommendation. Obtenido de https://www.w3.org/TR/rdf-syntax-grammar/

[29] W3C. (2004). OWL Web Ontology Language Reference. W3C Recommendation. Obtenido de https://www.w3.org/TR/owl-ref/

[30] Google. (s.f.). llaollao yogurt helado-ofertas. Obtenido de Google Play: https://play.google.com/store/apps/details?id=com.llaollao.app&hl=es

[31] Google. (s.f.). Compra online. Obtenido de Google Play: https://play.google.com/store/apps/details?id=com.bonpreu.mobile.android&hl=es

[32] Google. (s.f.). Pinterest. Obtenido de Google Play: https://play.google.com/store/apps/details?id=com.pinterest

[33] Google. (s.f.). Hacoo - Discovering &Inspiring. Obtenido de Google Play: https://play.google.com/store/apps/details?id=com.saramart.android

[34] Qué es la aplicación Hacoo, qué vende y por qué se ha hecho viral. (9 de Octubre de 2024). La Voz de Galicia. Obtenido de https://www.lavozdegalicia.es/noticia/reto-digital/ocio/2024/10/08/aplicacion-hacoo-vende-viral/00031728398212580172719.htm

[35] Google. (s.f.). YouTube Kids. Obtenido de Google Play: https://play.google.com/store/apps/details?id=com.google.android.apps.youtube.kids

[36] Google. (s.f.). Vlad and Niki – games & videos. Obtenido de Google Play: https://play.google.com/store/apps/details?id=me.apptivise.vladnikita

[37] Google. (s.f.). Twitch. Obtenido de Google Play: https://play.google.com/store/apps/details?id=tv.twitch.android.app

10. Anexos

Anexo A

A.1 Tabla

Categoría	
Ubicación	• Ubicación aproximada • Ubicación precisa
Información personal	• Nombre • Dirección de correo electrónico • ID de usuario • Dirección • Número de teléfono • Raza y etnia • Creencias políticas o religiosas • Orientación sexual • Otra información
Información financiera	• Información de pago del usuario • Historial de compras • Clasificación crediticia • Otra información financiera
Salud y fitness	• Información sanitaria • Información de estado físico
Mensajes	• Correos electrónicos • SMS o MMS • Otros mensajes desde apps
Fotos y videos	• Fotos • Videos
Archivos de audio	• Grabaciones de voz o de sonido • Archivos de música • Otros archivos de audio
Archivos y documentos	• Archivos y documentos
Calendario	• Eventos del calendario

Contactos	• Contactos
Actividad en apps	• Interacciones en la app • Historial de búsqueda en la app • Apps instaladas • Otro contenido generado por usuarios • Otras acciones
Navegación web	• Historial de navegación web
Información y rendimiento de la app	• Registros de fallas • Diagnóstico • Otros datos de rendimiento de apps
Dispositivo y otros ID	• Dispositivo u otros ID

TABLA 18: CATEGORÍAS PARA TIPO DE DATO

A.2 Diagramas para tipo de dato

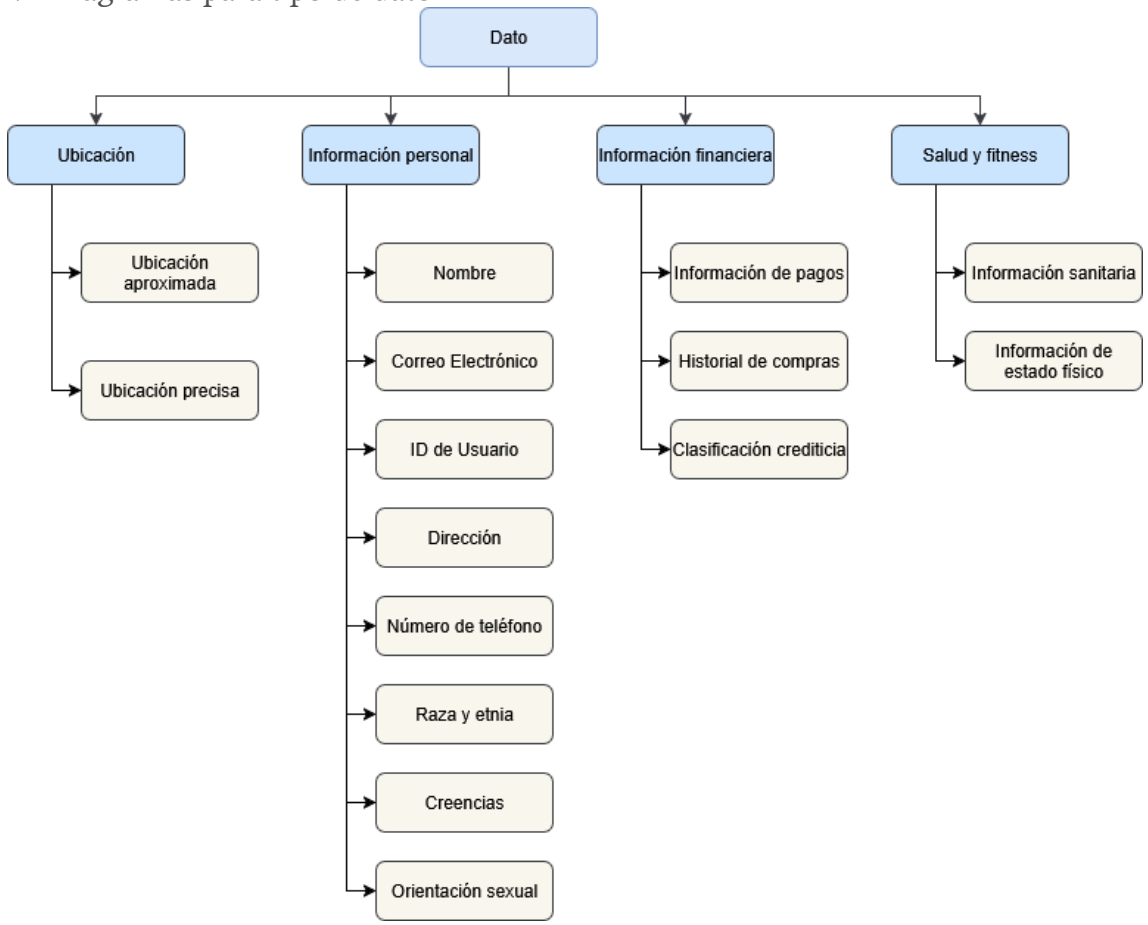

ILUSTRACIÓN 10: TIPO DE DATO (PRIMERA PARTE)

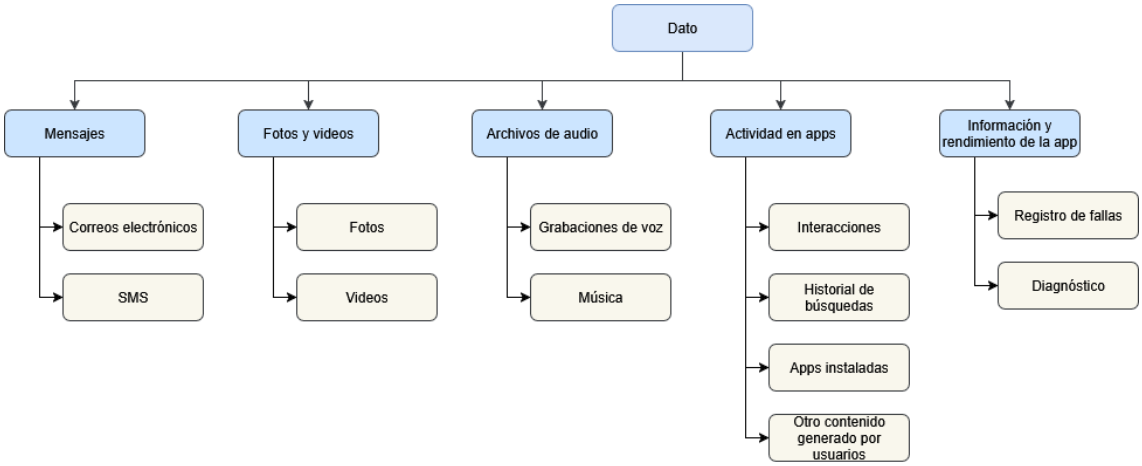

ILUSTRACIÓN 11: TIPO DE DATO (SEGUNDA PARTE)

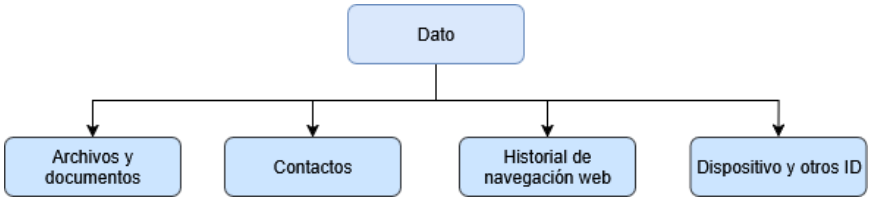

ILUSTRACIÓN 12: TIPO DE DATO (TERCERA PARTE)

Anexo B – URIs

Para las clases:

- http://tfg.uva.es/ontologias/datasecont#Aplicacion
- http://tfg.uva.es/ontologias/datasecont#TipoDeDato
- http://tfg.uva.es/ontologias/datasecont#Proposito
- http://tfg.uva.es/ontologias/datasecont#Comparticion
- http://tfg.uva.es/ontologias/datasecont#Manejo
- http://tfg.uva.es/ontologias/datasecont#Seguridad
- http://tfg.uva.es/ontologias/datasecont#Entidad

Subclases de TipoDeDato:

- http://tfg.uva.es/ontologias/datasecont#DatoPersonal
- http://tfg.uva.es/ontologias/datasecont#DatoFinanciero
- http://tfg.uva.es/ontologias/datasecont#DatoUbicacion
- http://tfg.uva.es/ontologias/datasecont#DatoSalud
- http://tfg.uva.es/ontologias/datasecont#DatoContacto
- http://tfg.uva.es/ontologias/datasecont#DatoMensaje
- http://tfg.uva.es/ontologias/datasecont#DatoMultimedia
- http://tfg.uva.es/ontologias/datasecont#ArchivosYDocumentos
  http://tfg.uva.es/ontologias/datasecont#DatoCalendario
- http://tfg.uva.es/ontologias/datasecont#DatoActividadApp
- http://tfg.uva.es/ontologias/datasecont#DatoNavegacionWeb
- http://tfg.uva.es/ontologias/datasecont#DatoRendimientoApp
- http://tfg.uva.es/ontologias/datasecont#DatoDispositivo

Tipos específicos de datos

- Dato Personal
  - http://tfg.uva.es/ontologias/datasecont#Nombre
  - http://tfg.uva.es/ontologias/datasecont#CorreoElectronico
  - http://tfg.uva.es/ontologias/datasecont#IDUsuario
  - http://tfg.uva.es/ontologias/datasecont#Direccion
  - http://tfg.uva.es/ontologias/datasecont#Telefono
  - http://tfg.uva.es/ontologias/datasecont#RazaOEtnia
  - http://tfg.uva.es/ontologias/datasecont#Creencias
  - http://tfg.uva.es/ontologias/datasecont#OrientacionSexual
  - http://tfg.uva.es/ontologias/datasecont#OtraInformacionPersonal
- Dato Financiero:
  - http://tfg.uva.es/ontologias/datasecont#PagoUsuario
  - http://tfg.uva.es/ontologias/datasecont#HistorialCompras
  - http://tfg.uva.es/ontologias/datasecont#ClasificacionCrediticia
  - http://tfg.uva.es/ontologias/datasecont#OtraInformacionFinanciera
- Dato Ubicación:
  - http://tfg.uva.es/ontologias/datasecont#UbicacionPrecisa
  - http://tfg.uva.es/ontologias/datasecont#UbicacionAproximada
- Dato Salud:
  - http://tfg.uva.es/ontologias/datasecont#InformacionSanitaria
  - http://tfg.uva.es/ontologias/datasecont#InformacionFisica

- Dato Mensaje:
    - http://tfg.uva.es/ontologias/datasecont#CorreoElectronico
    - http://tfg.uva.es/ontologias/datasecont#SMS
    - http://tfg.uva.es/ontologias/datasecont#MensajeApp
- Dato Multimedia:
    - http://tfg.uva.es/ontologias/datasecont#Foto
    - http://tfg.uva.es/ontologias/datasecont#Video
    - http://tfg.uva.es/ontologias/datasecont#Audio
    - http://tfg.uva.es/ontologias/datasecont#GrabacionVoz
    - http://tfg.uva.es/ontologias/datasecont#Musica
    - http://tfg.uva.es/ontologias/datasecont#OtroAudio
- Dato de Actividad App:
    - http://tfg.uva.es/ontologias/datasecont#Interacciones
    - http://tfg.uva.es/ontologias/datasecont#HistorialBusqueda
    - http://tfg.uva.es/ontologias/datasecont#AppsInstaladas
    - http://tfg.uva.es/ontologias/datasecont#ContenidoGeneradoPorUsuario
    - http://tfg.uva.es/ontologias/datasecont#OtrasAcciones
- Dato de Rendimiento App:
    - http://tfg.uva.es/ontologias/datasecont#RegistrosFalla
    - http://tfg.uva.es/ontologias/datasecont#Diagnostico
    - http://tfg.uva.es/ontologias/datasecont#OtroDatoRendimiento
- Dato de Archivo:
    - http://tfg.uva.es/ontologias/datasecont#ArchivosYDocumentos
- Dato de Calendario:
    - http://tfg.uva.es/ontologias/datasecont#EventoCalendario
- Dato de Navegación Web
    - http://tfg.uva.es/ontologias/datasecont#HistorialNavegacion
- Dato de Dispositivo
    - http://tfg.uva.es/ontologias/datasecont#IDDispositivo

Subclases de Entidad:

- http://tfg.uva.es/ontologias/datasecont#OrganizacionOrigen
- http://tfg.uva.es/ontologias/datasecont#Tercero

Subclases de Seguridad:

- http://tfg.uva.es/ontologias/datasecont#CifradoEnTransito
- http://tfg.uva.es/ontologias/datasecont#RevisadoMAVS
- http://tfg.uva.es/ontologias/datasecont#PermiteEliminacion
- http://tfg.uva.es/ontologias/datasecont#NoCifrado
- http://tfg.uva.es/ontologias/datasecont#NoPermiteEliminacion
- http://tfg.uva.es/ontologias/datasecont#CumplePoliticaFamilias

Jerarquía de Propósitos:

- http://tfg.uva.es/ontologias/datasecont#PropositoRecopilacion
    - http://tfg.uva.es/ontologias/datasecont#PropositoAdministrativo
    - http://tfg.uva.es/ontologias/datasecont#AdministracionDeLaCuenta
    - http://tfg.uva.es/ontologias/datasecont#PropositoFuncional
        - http://tfg.uva.es/ontologias/datasecont#FuncionesDeLaApp

        - http://tfg.uva.es/ontologias/datasecont#ComunicacionesDelDesarrollador
    - http://tfg.uva.es/ontologias/datasecont#PropositoAnalitico
        - http://tfg.uva.es/ontologias/datasecont#Analisis
        - http://tfg.uva.es/ontologias/datasecont#Personalizacion
    - http://tfg.uva.es/ontologias/datasecont#PropositoLegal
        - http://tfg.uva.es/ontologias/datasecont#SeguridadCumplimientoYPrevencion
    - http://tfg.uva.es/ontologias/datasecont#PropositoComercial
        - http://tfg.uva.es/ontologias/datasecont#PublicidadOMarketing
- http://tfg.uva.es/ontologias/datasecont#PropositoComparticion
    - http://tfg.uva.es/ontologias/datasecont#PropositoComparticionTercero
    - http://tfg.uva.es/ontologias/datasecont#PropositoComparticionProveedorServicios
    - http://tfg.uva.es/ontologias/datasecont#PropositoComparticionGubernamental
    - http://tfg.uva.es/ontologias/datasecont#PropositoAnonimizacion

Propiedades de objeto:

- http://tfg.uva.es/ontologias/datasecont#recopilaDato
- http://tfg.uva.es/ontologias/datasecont#comparteCon
- http://tfg.uva.es/ontologias/datasecont#esRequeridoPara
- http://tfg.uva.es/ontologias/datasecont#seCompartePara
- http://tfg.uva.es/ontologias/datasecont#usaEncriptacion
- http://tfg.uva.es/ontologias/datasecont#implementaMedidaSeguridad
- http://tfg.uva.es/ontologias/datasecont#esPropositoDeComparticion
- http://tfg.uva.es/ontologias/datasecont#esPropositoDeRecopilacion
- http://tfg.uva.es/ontologias/datasecont#permiteEliminacionDatos

Propiedades de dato:

- http://tfg.uva.es/ontologias/datasecont#datoEsSensitivo
- http://tfg.uva.es/ontologias/datasecont#modoComparticion
- http://tfg.uva.es/ontologias/datasecont#procesamientoEfimero
- http://tfg.uva.es/ontologias/datasecont#datoOpcional
- http://tfg.uva.es/ontologias/datasecont#modoUsoDato

Para los individuos, planteamos los siguientes ejemplos de instancias:

- http://tfg.uva.es/ontologias/datasecont#com_llaollao_app
- http://tfg.uva.es/ontologias/datasecont#com_bonpreu_mobile_android
- http://tfg.uva.es/ontologias/datasecont#com_pinterest
- http://tfg.uva.es/ontologias/datasecont#com_saramart_android
- http://tfg.uva.es/ontologias/datasecont#com_google_android_apps_youtube_kids
- http://tfg.uva.es/ontologias/datasecont#me_apptivise_vladnikita
- http://tfg.uva.es/ontologias/datasecont#tv_twitch_android_app

Anexo C – Código del TFG

URL al repositorio: https://gitlab.inf.uva.es/alegavi/datasecont

Contenido:

- Ontología
  - Ontología v1 (DataSecOnt_v1.owl)
  - Ontología v2 (DataSecOnt_v2.owl)
  - Ontología v3 (DataSecOnt_v3.owl)
  - Ontología v4 (DataSecOnt_v4.owl)
  - Ontología v5 (DataSecOnt_v5.owl)
- Resultados
  - Resultados de las CQs (query-result_1_.csv, …, query-result_15_.csv)
  - Resultados de las consultas extra (extra-result_1_.csv, …, extra-result_10_.csv)

Anexo D – Proceso de implementación

A continuación se documenta, de forma estructurada, el proceso seguido para implementar la ontología DataSecOnt en un entorno funcional, capaz de soportar consultas SPARQL y razonamiento semántico. Este proceso incluyó tanto la configuración del entorno técnico como la carga efectiva del modelo RDF/OWL en la plataforma GraphDB.

1. Entorno técnico: máquina virtual y acceso remoto

Para poder trabajar con GraphDB en un entorno gráfico completo, se habilitó una máquina virtual Ubuntu 22.04.5. Esta máquina sirvió como entorno aislado y persistente para la instalación y uso de GraphDB, así como para otras tareas auxiliares como la carga de ficheros RDF o la depuración de consultas SPARQL.

1.1. Instalación de Ubuntu Desktop (servicio SLIM)

Dado que la versión base de la máquina virtual no incluía entorno gráfico, se optó por instalar un entorno de escritorio ligero mediante el paquete ubuntu-desktop-minimal, compatible con el servicio SLIM. Este entorno permitió ejecutar navegadores gráficos y acceder cómodamente a la interfaz de GraphDB.

1.2. Habilitación del acceso remoto con Google Remote Desktop

Para acceder a la máquina virtual de forma visual desde cualquier equipo, se utilizó Google Remote Desktop, una herramienta gratuita que permite conectarse al entorno gráfico de una máquina remota mediante navegador web.

El proceso incluyó:

- La instalación del paquete chrome-remote-desktop en la máquina virtual.
- La vinculación de la máquina a una cuenta de Google para autorizar el acceso remoto.
- La configuración del entorno de escritorio como sesión predeterminada para el servicio remoto.

Una vez configurado, fue posible conectarse gráficamente a la máquina virtual desde cualquier navegador mediante la consola de Google Remote Desktop, lo que permitió abrir GraphDB en el navegador local (http://localhost:7200) y trabajar de forma fluida.

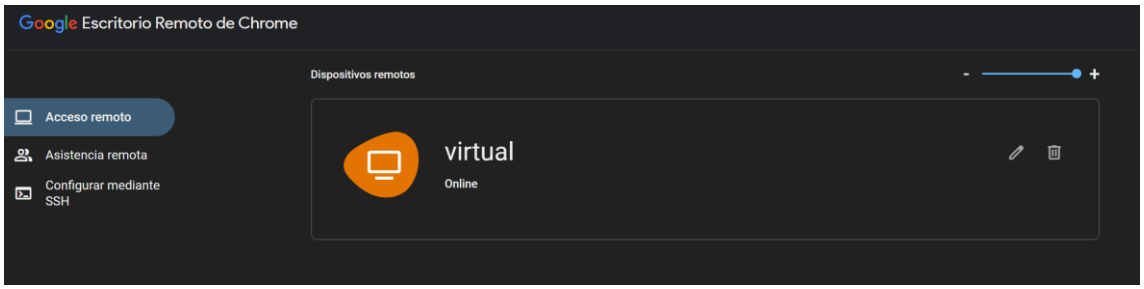

ILUSTRACIÓN 13: ESCRITORIO REMOTO

2. Creación del repositorio en GraphDB

El primer paso fue la creación de un repositorio dentro de GraphDB. Se generó un nuevo repositorio denominado datasecont_repo, configurado específicamente con soporte para el perfil OWL2 RL. Este perfil de razonamiento es el más adecuado cuando se busca un equilibrio entre capacidad inferencial y eficiencia computacional, ya que permite:

- Razonamiento sobre jerarquías de clases (rdfs:subClassOf).
- Validación de restricciones lógicas como dominios, rangos y cardinalidades.
- Inferencia sobre propiedades equivalentes y disjuntas.

Una vez creado, el repositorio quedó accesible a través de la interfaz web de GraphDB en la URL:

http://virtual:7200/repositories/datasecont_repo

```
graphdb-10.8.1
usuario@virtual:~/graphdb$ cd graphdb-10.8.1/
usuario@virtual:~/graphdb/graphdb-10.8.1$ ls
bin  conf  configs  data  doc  examples  lib  logs  README  tools  work
usuario@virtual:~/graphdb/graphdb-10.8.1$ cd bin/
usuario@virtual:~/graphdb/graphdb-10.8.1/bin$ ls
cluster-proxy        graphdb-check.py      rdfvalidator
cluster-proxy.cmd    graphdb.cmd           rdfvalidator.cmd
console              graphdb.in.cmd        reification-convert
console.cmd          graphdb.in.sh         reification-convert.cmd
convert-repo         graphdb-java.in.cmd   rule-compiler
convert-repo.cmd     ibrtool               rule-compiler.cmd
env-vars.in.cmd      ibrtool.cmd           setvars.in.cmd
env-vars.in.sh       importrdf             setvars.in.sh
generate-report      importrdf.cmd         storage-tool
generate-report.cmd  migration-wizard      storage-tool.cmd
graphdb              migration-wizard.cmd
usuario@virtual:~/graphdb/graphdb-10.8.1/bin$ ./graphdb
./graphdb: line 75: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
[INFO ] 2025-05-01 16:19:33,291 [main | c.o.g.Config] Using 'file:/home/usuario/graphdb/graph
db-10.8.1/conf/logback.xml' as logback's configuration file
[INFO ] 2025-05-01 16:19:33,827 [main | c.o.g.s.GraphDB] Starting GraphDB in workbench mode.
S[INFO ] 2025-05-01 16:19:53,942 [main | c.o.g.Config] GraphDB Home directory: /home/usuario/
graphdb/graphdb-10.8.1
[INFO ] 2025-05-01 16:19:53,943 [main | c.o.g.Config] GraphDB Config directory: /home/usuario
/graphdb/graphdb-10.8.1/conf
[INFO ] 2025-05-01 16:19:53,943 [main | c.o.g.Config] GraphDB Data directory: /home/usuario/g
raphdb/graphdb-10.8.1/data
[INFO ] 2025-05-01 16:19:53,944 [main | c.o.g.Config] GraphDB Work directory: /home/usuario/g
raphdb/graphdb-10.8.1/work
[INFO ] 2025-05-01 16:19:53,944 [main | c.o.g.Config] GraphDB Logs directory: /home/usuario/g
raphdb/graphdb-10.8.1/logs
```

ILUSTRACIÓN 14: CONEXIÓN A GRAPHDB

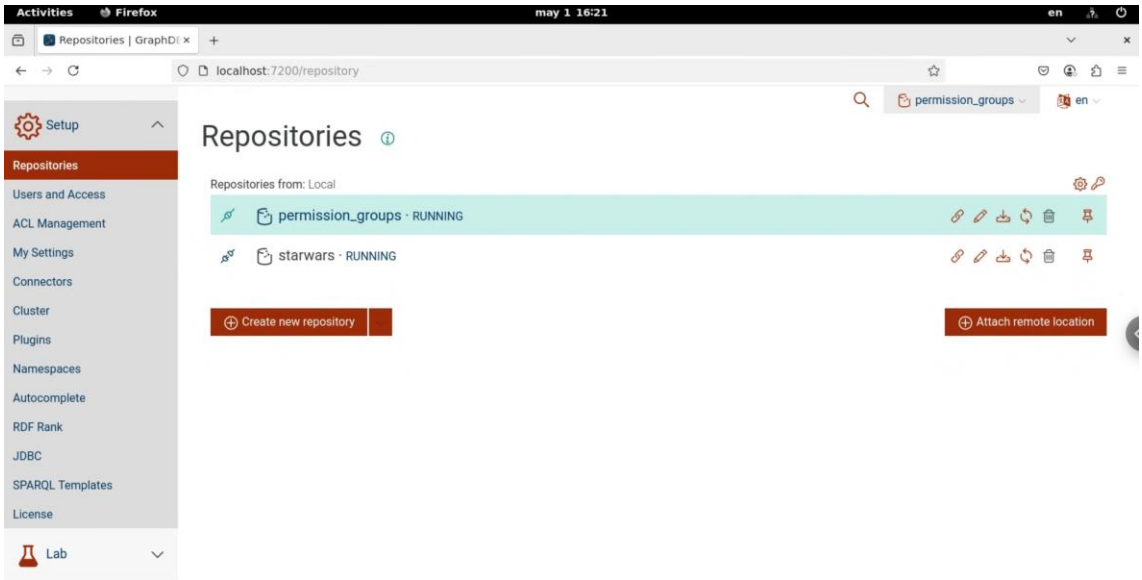

ILUSTRACIÓN 15: REPOSITORIOS EN GRAPHDB

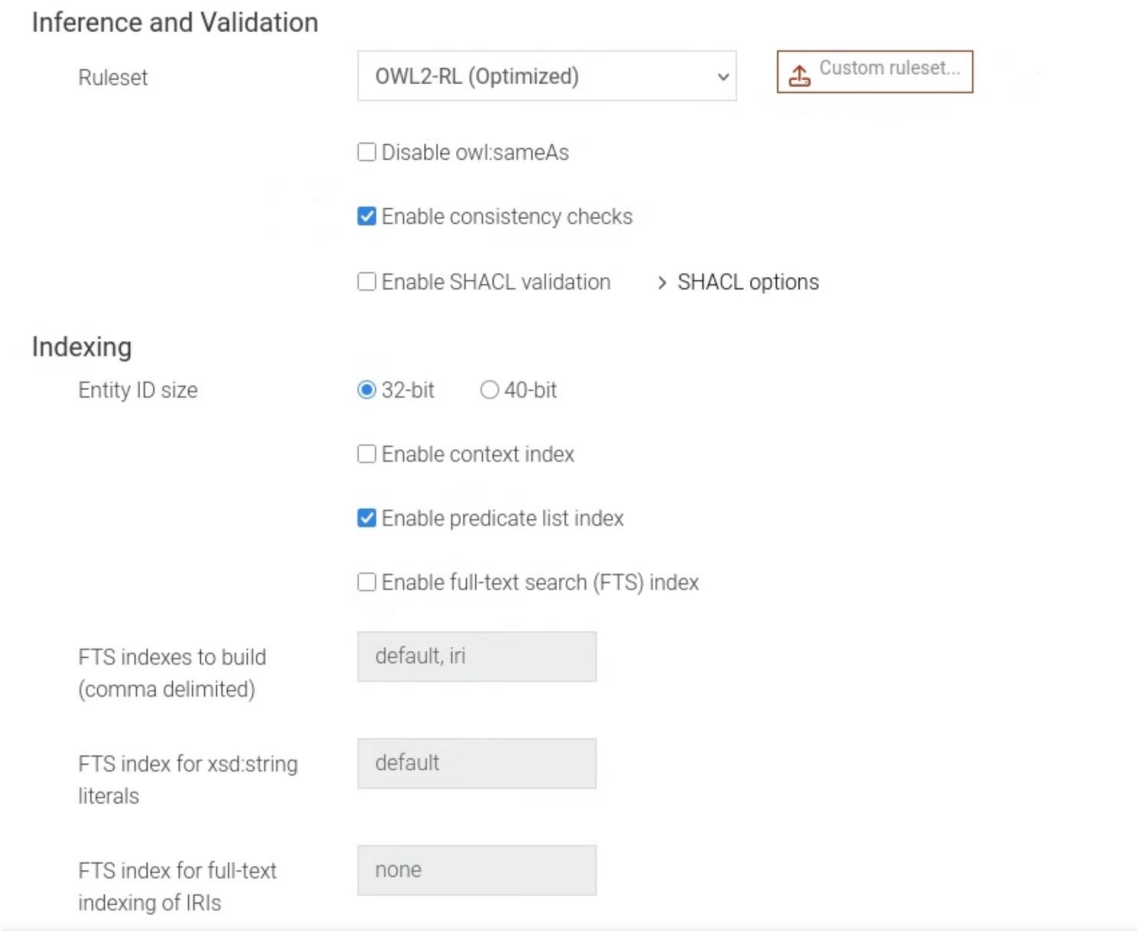

ILUSTRACIÓN 16: CONFIGURACIÓN DEL REPOSITORIO EN GRAPHDB

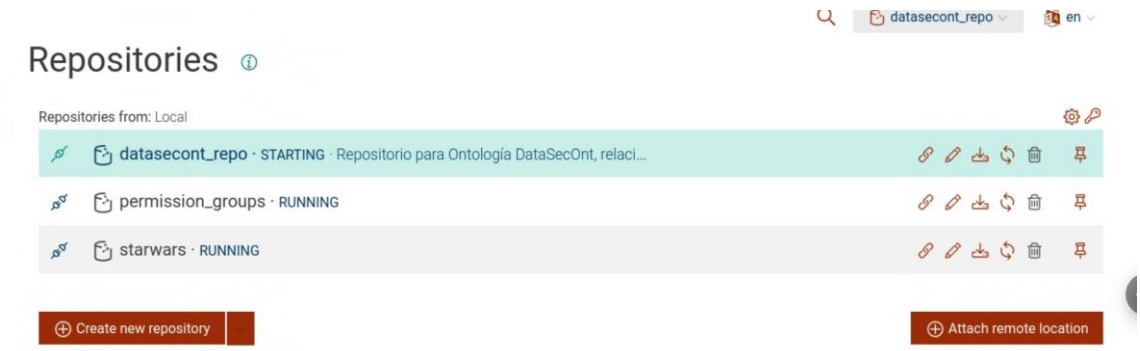

ILUSTRACIÓN 17: CREACIÓN DE REPOSITORIO EN GRAPHDB

3. Desarrollo del archivo OWL de la ontología

El archivo DataSecOnt_v1.owl, que contiene la representación formal de la ontología, ha sido desarrollado íntegramente por la autora del presente trabajo utilizando el editor de código Visual Studio Code. Este entorno permitió trabajar directamente con la sintaxis RDF/XML, permitiendo un control preciso sobre cada elemento del modelo ontológico y favoreciendo una comprensión profunda de la estructura semántica definida.

El desarrollo se realizó siguiendo los principios de la Web Semántica y aplicando la metodología *Ontology Development 101* como guía para la identificación de conceptos clave, relaciones y restricciones. En este proceso se definieron manualmente:

- Las clases y subclases que estructuran el dominio, como Aplicacion, TipoDeDato, Finalidad, PrácticaDeSeguridad, entre otras.

- Las propiedades de objeto y de datos, con sus respectivos dominios y rangos.

- Las restricciones lógicas expresadas mediante OWL, tales como cardinalidades mínimas y máximas, equivalencias entre clases o relaciones de disjunción.

- Anotaciones semánticas (rdfs:label, rdfs:comment) para documentar el modelo y facilitar su lectura por parte de otros usuarios.

Durante el desarrollo, se validó sintácticamente el archivo con extensiones de validación XML y RDF disponibles en Visual Studio Code, y se realizaron comprobaciones progresivas para asegurar la consistencia estructural antes de su carga en GraphDB. Esta forma de desarrollo manual permitió un alto nivel de personalización y precisión en la representación del conocimiento.

4. Carga del archivo RDF/XML

A través del módulo de importación de GraphDB, se cargó el archivo DataSecOnt_v1.owl, que contiene la definición completa de la ontología desarrollada. Este archivo fue exportado previamente desde la herramienta Protégé en formato RDF/XML, una de las serializaciones más ampliamente soportadas por los sistemas de almacenamiento semántico.

Durante el proceso de carga, el sistema reconoció correctamente todas las clases, propiedades y axiomas definidos, incluyendo:

- La jerarquía de clases (rdfs:subClassOf)

- Las propiedades de objeto y de datos, junto con sus dominios y rangos

- Las restricciones lógicas definidas mediante OWL, como cardinalidades mínimas o relaciones de disjunción

- Las anotaciones descriptivas (rdfs:label, rdfs:comment) utilizadas para mejorar la legibilidad semántica del modelo

Una vez completada la importación, GraphDB activó automáticamente el razonador OWL2 RL, configurado previamente en el repositorio datasecont_repo. Este razonador permitió inferir nuevo conocimiento a partir de las reglas declaradas en la ontología, como la pertenencia de instancias a clases más generales o el cumplimiento de propiedades implícitas a través de la jerarquía.

La correcta activación del razonador y la validación de los axiomas cargados confirmaron que la ontología era consistente y semánticamente operativa. Además, la interfaz Workbench permitió explorar visualmente la estructura de la ontología y ejecutar las primeras consultas SPARQL de prueba, comprobando que las inferencias se realizaban correctamente y que los resultados respondían a las preguntas de competencia previamente definidas.

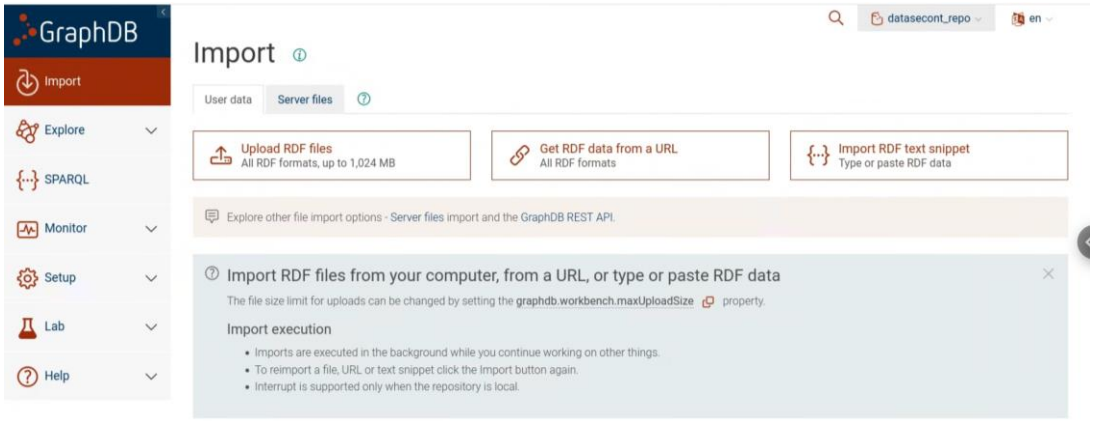

ILUSTRACIÓN 18: IMPORTACIÓN EN GRAPHDB

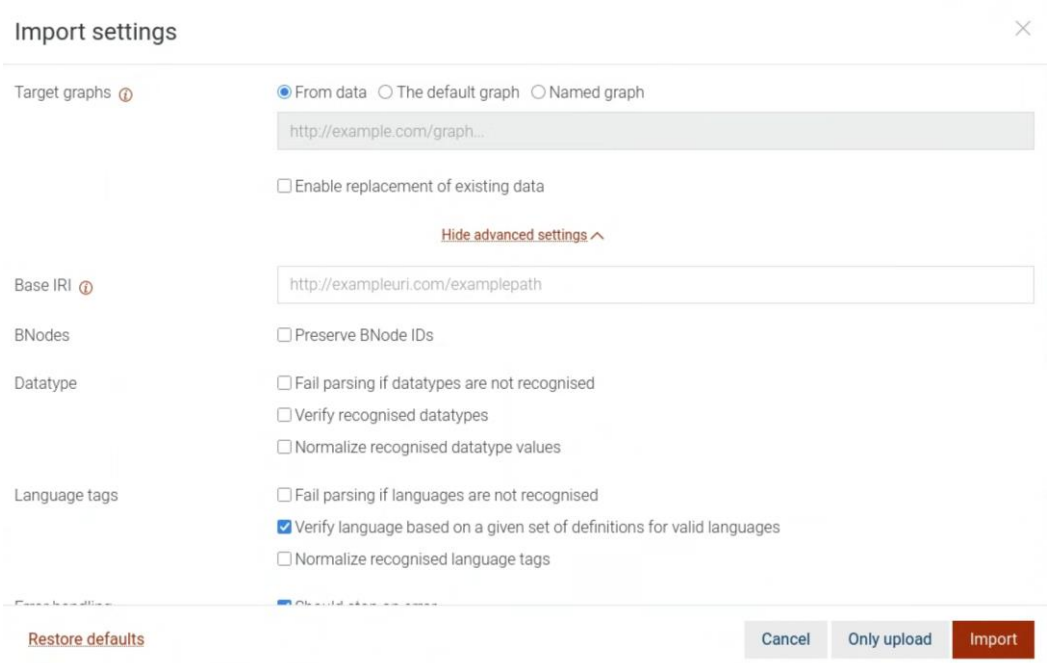

ILUSTRACIÓN 19: CONFIGURACIÓN PARA IMPORTACIÓN EN GRAPHDB

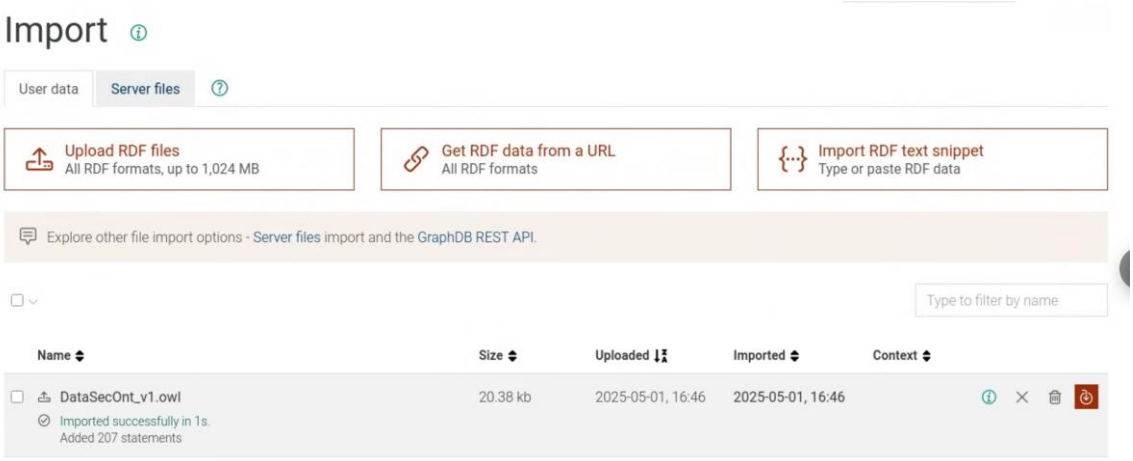

ILUSTRACIÓN 20: RESULTADO IMPORTACIÓN EN GRAPHDB

Anexo E – Visual graphs

La red principal de TipoDeDato:

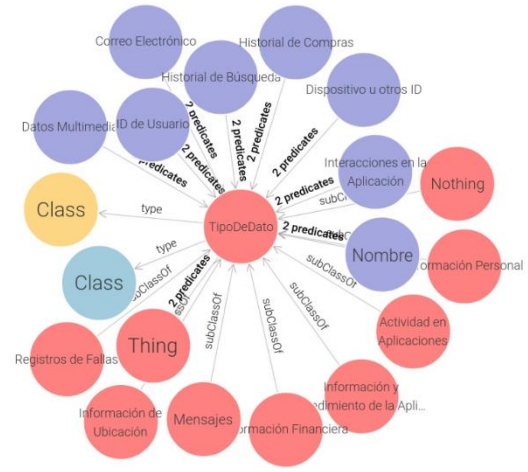

ILUSTRACIÓN 21: RED DE TIPO DE DATO

TipoDeDato expandido junto a los nodos correspondientes de Información Personal:

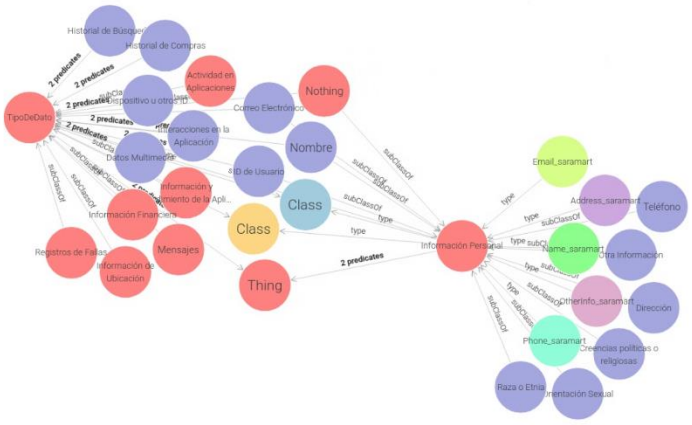

ILUSTRACIÓN 22: RED DE TIPO DE DATO E INFORMACIÓN PERSONAL

Detalle de TipoDeDato e Interacciones en la Aplicación:

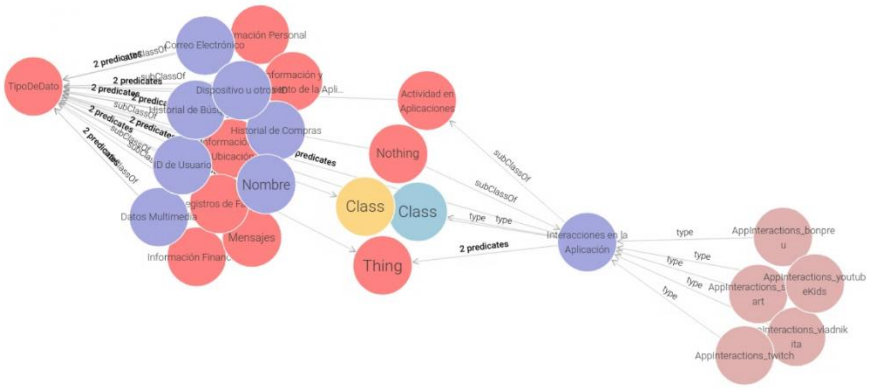

ILUSTRACIÓN 23: RED DE TIPO DE DATO E INTERACCIONES EN LA APLICACIÓN