Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Servicio de web semántica a través de una base de conocimiento para compañeros digitales enfocado a pacientes con diabetes

TESIS

que presenta

Ing. Yanitza de la Caridad Gutiérrez Acea

para obtener el Grado de

Maestra en Ciencias

en Computación

Directora de Tesis

Dra. Sonia Guadalupe Mendoza Chapa

Ciudad de México **Noviembre de 2024**

Índice general

Resumen

El campo de la salud digital está experimentado un crecimiento exponencial, con un enfoque cada vez mayor en la integración de datos y la creación de sistemas inteligentes que ayuden a los pacientes, médicos y proveedores de atención médica en la búsqueda de información y la toma de decisiones. La relación entre la web semántica y las bases de conocimiento, en este contexto, es un elemento esencial hacia la consecución de esta meta. Actualmente, se cuenta con trabajos que plantean la integración e implementan ambas tecnologías, pero no todos tienen como dominio o área de aplicación a la salud o se especializan en enfermedades crónicas no transmisibles (ENT) como la diabetes. Este proyecto se centra en la convergencia de dos poderosas tecnologías: la web semántica y las bases de conocimiento. Se implementa un microservicio bajo la especificación de *Restful Web Service*, mediante el desarrollo de un servicio de web semántica utilizando una base de conocimiento para apoyar el proceso de búsqueda semántica. Ambas tecnologías se integran de manera que exista una conexión mutua durante el procedimiento de consultas o búsqueda de información asociada a temas de salud, específicamente de diabetes. Para ello, nos apoyamos en herramientas como Visual Paradigm para el modelado de los datos, PostgreSQL y PgAdmin como herramientas de gestión y administración de la base de datos respectivamente, Pentaho Data Integration para el proceso de extracción, transformación y carga (ETL) de los datos, Protégé para la confección de la ontología y Python como lenguaje de programación para la implementación del servicio de web semántica, apoyado de *Flask* como *framework* de desarrollo. La base de conocimiento es desarrollada a partir de un mercado de datos, donde su implementación está basada en el enfoque de Kimball. A través de la identificación de las tablas de hechos y dimensiones, se definieron los principales conceptos, clases o tópicos del dominio, que permitieron la construcción de la ontología. A partir de esta, se creó el grafo de conocimiento que guía el proceso de búsqueda tanto en la Web, mediante la técnica de *Scraping*, como en la base de conocimiento a través de consultas SPARQL. Para el servicio de web semántica se propone la arquitectura, sus componentes y el proceso de implementación de la misma. Las pruebas del prototipo con usuarios finales se realzaron mediante del cuestionario NASA-TLX (*NASA Task Load Index*), para medir la carga de trabajo de la solución propuesta y el cuestionario de *AttrakDiff* para medir la percepción y usabilidad estética que los usuarios perciben al utilizar la aplicación. Estas pruebas arrojaron resultados positivos, ya que el producto resultó deseado por los usuarios y con una carga de trabajo no significativa.

Palabras clave: base de conocimiento, compañeros digitales, mercado de datos, diabetes, enfermedades crónicas no transmisibles (ENT), *Restful Web Service*, semántica.

Abstract

The field of digital health is experiencing exponential growth, with an increasing focus on data integration and the creation of intelligent systems that assist patients, doctors, and healthcare providers in searching for information and making decisions. The relationship between the semantic web and knowledge bases, in this context, is an essential element towards achieving this goal. However, while there are existing works that propose and implement the integration of both technologies, not all focus on or specialize in health or chronic non-communicable diseases (NCDs) such as diabetes. This project focuses on the convergence of two powerful technologies: the semantic web and knowledge bases. A microservice is implemented based on the Restful Web Service specification, through the development of a semantic web service utilizing a knowledge base to support the semantic search process. Both technologies are integrated to establish a mutual connection during the querying or information retrieval process related to health topics, specifically diabetes. To achieve this, we rely on tools such as Visual Paradigm for data modeling, PostgreSQL and PgAdmin for database management and administration respectively, Pentaho Data Integration for the process of data extraction, transformation, and loading (ETL), Protégé for ontology creation, and Python as the programming language for the implementation of the semantic web service, supported by Flask as the development framework. The knowledge base is developed from a data marketplace, where its implementation is based on the Kimball approach. Through the identification of fact and dimension tables, the main concepts, classes, or topics of the domain were defined, which enabled the construction of the ontology. From it, we created the knowledge graph that guides the search process both on the web through the *scraping* technique and within the knowledge base through SPARQL queries. The architecture, components, and implementation process of the semantic web service are proposed in this research project. The tests of the prototype with end users were conducted using the NASA-TLX (*NASA Task Load Index*) questionnaire to measure the workload of the proposed solution and the *AttrakDiff* questionnaire to assess the aesthetic perception and usability that users experience when using the application. These tests yielded positive results, as the product was found to be desirable by the users and with non-significant workload.

Keywords: Knowledge Base, Digital Companions, Data Mart, Diabetes, Non-communicable Chronic Diseases (NCDs), Restful Web Service, Semantic Web.

Agradecimientos

Cuando la gratitud es absoluta, las palabras sobran, pero en silencio no sirve a nadie. Es por ello que quiero agradecer:

Primeramente, a mi hijo Ian Lucas, quien, a pesar de la distancia que nos separa, sigue siendo mi mayor motivación y la razón detrás de cada paso en este camino. Aunque esté lejos, cada pensamiento en ti me impulsa a seguir adelante, a dar lo mejor de mí en esta meta. Gracias, hijo, por ser mi inspiración diaria y recordarme que todo esfuerzo vale la pena, por tu paciencia y comprensión. Esta tesis es también un pedacito de lo que construyo para ti, con la esperanza de ser alguien de quien puedas sentirte orgulloso algún día.

A mis padres, Juana y Rolando, cuyo amor y apoyo incondicional han sido el pilar más firme en mi vida. Gracias por enseñarme el valor del esfuerzo y la perseverancia, y por acompañarme en cada etapa de este camino, con paciencia y fe en mí, incluso en los momentos de duda. Su ejemplo de dedicación y sacrificio es mi guía y sin su confianza en mis sueños, este logro no habría sido posible. A ustedes, que siempre han creído en mí y me han alentado a ir más allá, les debo esta meta alcanzada y mucho más.

A mis hermanos por su constante preocupación y apoyo en obtener mis logros, sin importar la distancia o las circunstancias. Gracias por ser ese respaldo en los momentos de mayor desafío. Al resto de mi familia, por cada gesto de aliento que me ha motivado a seguir adelante, celebrar mis logros como propios y por cada palabra de ánimo y consuelo. Cada uno de ustedes han sido una fuente de fortaleza y alegría, recordándome siempre de dónde vengo y por quiénes sigo adelante.

A la nueva familia que adquirí en México, quienes han sido mi refugio y fortaleza en este país que ahora también es mi hogar. Cuando uno deja su tierra en busca de un futuro mejor y se embarca en camino de alcanzar sus sueños, encontrar personas que te brindan su apoyo incondicional es un regalo invaluable. A mis hermanos cubanos, por compartir nuestras raíces, nuestras historias y por darme ese calor de hogar que tanto se extraña. A mis amigos mexicanos, gracias por acogerme con los brazos abiertos, por cada muestra de cariño y por hacerme sentir como en casa en esta hermosa tierra.

Agradezco además, de manera especial, al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), por financiar gratuitamente los dos años de estudios de maestría. Al Centro de Investigación y de Estudios Avanzados (CINVESTAV) y al Departamento de Computación, por ser un espacio de crecimiento, aprendizaje y constante inspiración. Gracias por ofrecerme las herramientas, el entorno y el respaldo necesarios para avanzar en mis estudios y perseguir mis metas académicas.

Por último, pero no menos importante, quiero expresar mi más sincero agradecimiento a la Dra. Sonia Guadalupe Mendoza Chapa, mi asesora de tesis, por su guía, paciencia y dedicación. Su compromiso y generosidad al compartir sus conocimientos han sido esenciales en mi formación. Gracias por cada consejo, por sus palabras de aliento y por exigirme siempre dar lo mejor de mí.

A todos, gracias por ser parte de esta historia.

Capítulo 1

Introducción

La diabetes *mellitus* (en adelante «diabetes» para simplificar) es una de las enfermedades no transmisibles (ENT) que más afecta a la población mundial. Es una enfermedad metabólica crónica que se caracteriza por elevados niveles de glucosa (azúcar) en la sangre. Se considera como la sexta causa de mortalidad en América y la tercera en México [41] [23].

La OPS (Organización Panamericana de la Salud) en conjunto con la OMS (Organización Mundial de la Salud) adoptaron una serie de medidas con el objetivo de estimular y apoyar la prevención y control de la diabetes, así como las complicaciones que esta puede causar. Estas medidas se basan en: [41]

- Enfatizar en la vigilancia de la diabetes y factores de riesgo.
- Desarrollar normas y estándares para el diagnóstico y la atención de la diabetes.
- Facilitar directrices científicas que garanticen la prevención de la diabetes.

México no está ajeno a estas medidas, pues trabaja a nivel de salud para controlar y prevenir la diabetes. El desarrollo tecnológico, con el que cuenta el país, hace posible que se tenga mayor control de dicha enfermedad y los riesgos asociados a ella, pues en este aspecto se utilizan las TIC (Tecnologías de la Información y las Comunicaciones) para el manejo estadístico de datos y cifras significativas de la diabetes, así como su diagnóstico. Además, estos avances incluyen tecnologías pervasivas, como compañeros digitales en diferentes ámbitos, con el objetivo de ayudar a los expertos del área durante el análisis, el control y la gestión del proceso para el que fue creado [59]. Asimismo, a nivel individual, los compañeros digitales apoyan a las personas con la realización de tareas que a menudo se tornan rutinarias o requieran mucho tiempo [20].

En el ámbito de la salud, un compañero digital debe ser capaz de interactuar con el paciente y proporcionarle información relevante sobre su enfermedad, así como monitorear de manera constante las actividades de medicación, alimentación, programación de citas, etc.

En este contexto, la incorporación de tecnologías como servicios de web semántica y bases de conocimiento, son de vital importancia teniendo en cuenta aspectos como la optimización, la integración, el rendimiento de los compañeros digitales para la gestión y el control de la diabetes, en cuanto a la mejora de la calidad de vida de los pacientes. La integración de la web semántica y las bases de conocimiento en la gestión de la diabetes a través de compañeros digitales ofrece beneficios como:

- **Mejora en la toma de decisiones**: el acceso a datos integrados y contextualizados permite tomar decisiones más claras y deterministas, tanto para los profesionales de la salud como para el paciente. Esto se traduce en un manejo más efectivo de la diabetes, ya que se pueden ajustar el tratamiento y las recomendaciones, de manera precisa, según los datos específicos del paciente.

- **Automatización inteligente**: el uso de compañeros digitales proporcionan la automatización de tareas, recordatorios personalizados sobre la medicación, dietas y citas médicas. Esta automatización no sólo reduce la carga cognitiva del paciente, sino que también garantiza que no se omitan aspectos importantes del manejo de la enfermedad.
- **Interacción avanzada**: los pacientes pueden interactuar con los compañeros digitales de forma natural, obteniendo respuestas precisas a sus búsquedas. La capacidad de comprensión y procesamiento del lenguaje natural permite que el compañero digital actúe como un asistente confiable y accesible, proporcionando información en tiempo real y aclarando dudas que el paciente pueda tener.

En esta tesis, se explora cómo un servicio de web semántica, a través de una base de conocimiento, ayuda a optimizar el funcionamiento de los compañeros digitales. Este enfoque no sólo mejora la gestión y el control de la enfermedad, sino que también promueve una atención más integral y personalizada, adaptada a las necesidades específicas de cada paciente.

1.1 Antecedentes

El presente trabajo de tesis se inscribe en el proyecto doctoral de Guillermo Monroy Rodríguez que lleva por título «Ambiente computacional para compañeros digitales orientado al cuidado de la salud empleando infraestructura pervasiva», que es una propuesta de arquitectura para compañeros digitales en el área de la salud. Dicha arquitectura cuenta con ocho componentes o microservicios (ver Figura 1.1): 1) Reconocimiento de Voz, 2) Inteligencia del Comportamiento, 3) Monitoreo de Datos Biométricos, 4) Servicio de Web Semántica, 5) Servicio de Respuestas Proactivas, 6) Base de Conocimiento, 7) Broker y 8) Compañero Digital, cada uno de los cuales presentan funcionalidades específicas y bien definidas.

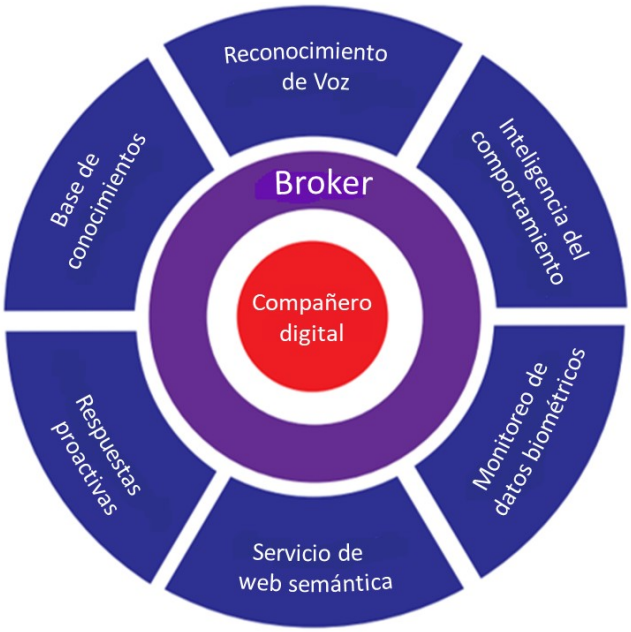


Figura 1.1: Vista de servicios de la arquitectura para compañeros digitales en el ámbito de la salud.

1. **Reconocimiento de Voz**: este componente recibe señales de audio y realiza acciones ordenadas por los usuarios a través de la identificación de palabras claves.

2. **Inteligencia del Comportamiento**: una vez obtenidos los datos recibidos de los componentes **Monitoreo de Datos Biométricos** y **Compañero Digital**, este componente se encarga del entrenamiento de los modelos de aprendizaje automático o computacional y la ejecución de modelos de agrupamiento y clasificación, haciendo uso de dispositivos de *Edge Artificial Intelligence*.

3. **Monitoreo de Datos Biométricos**: la función de este componente radica en el monitoreo de patrones biométricos, así como su recolección y estructuración para su posterior análisis.

4. **Servicio de Web Semántica**: este servicio se basa en la realización de consultas que pueden hacerse directamente al componente **Base de Conocimiento** o, en caso de no existir o no encontrar respuestas en este componente, se realiza la búsqueda en la Web y, al mismo tiempo, habrá una retroalimentación de la Base de Conocimiento para futuras consultas similares.

5. **Servicio de Respuestas Proactivas**: este componente interactúa con el usuario mediante el envío de respuestas a las solicitudes del usuario, así como información y sugerencias proactivas. Esto permite que el usuario mantenga el control de su estado de salud.

6. **Base de Conocimiento**: este componente se fundamenta en una base de conocimiento que contiene información, que luego será consultada a través del componente de **Web Semántica**. Al mismo tiempo, adquiere nuevo conocimiento, ya que, en caso de no encontrarse la respuesta a una consulta, esta se realiza a través de la Web.

7. **Broker**: este componente sirve de intermediario entre todos los componentes, ya que se encarga del envío y recepción de las peticiones entre los diferentes componentes.

8. **Compañero Digital**: este componente contiene la lógica del negocio en cuanto a la gestión de citas, seguimiento médico, etc. Se encarga, además, de la gestión de la información y del conocimiento existente, para luego generar un modelo de retroalimentación que adapte el compañero digital a las necesidades del usuario.

1.2 Planteamiento del problema

Existen proyectos que manejan bases de conocimientos, las cuales proporcionan información para el sector de la salud, pero la mayoría se basa en recopilar la información a través de ontologías, registros médicos electrónicos, bases de datos públicas o repositorios. Sin embargo, estas iniciativas carecen de una integración holística entre bases de conocimientos y servicios de web semántica, en la que cada una se nutra de la otra. Además, una vez conformada la base de conocimiento, esta sirva de fuente primaria para las consultas a través de la web semántica y, al mismo tiempo, se enriquezca con conocimiento nuevo generado de las búsquedas semánticas enfocadas a pacientes de diabetes, para su autocontrol y gestión de su medicación, tratamiento y forma de vida. Esta desconexión no sólo limita la eficacia de las búsquedas informativas, y la profundidad y relevancia de

las respuestas proporcionadas a las consultas de los usuarios, sino que también restringe la posibilidad de un potenciamiento mutuo entre estas tecnologías, lo cual es vital para una gestión de la salud más dinámica y personalizada.

Además, se dificulta la consolidación y accesibilidad de la información de manera coherente y contextualizada. Los usuarios, ya sean pacientes o profesionales de la salud, pueden enfrentarse a respuestas estáticas y limitadas. La automatización de tareas clave, como el envío de recordatorios de medicación o la gestión de citas médicas, se ve obstaculizada al no haber un sistema de integración. Los sistemas de salud digital enfrentan desafíos significativos, en términos de adopción y escalabilidad, cuando no pueden aprovechar plenamente las capacidades integradas de estas tecnologías.

En este contexto, surge la necesidad de mejorar la eficiencia y efectividad de los compañeros digitales mediante el uso de servicios de web semántica y bases de conocimiento. Esta integración es esencial para superar las limitaciones actuales y avanzar hacia soluciones más eficaces y centradas en el paciente en el campo de la salud digital.

1.3 Hipótesis

La integración de una base de conocimiento a un servicio de web semántica ofrecerá, además de conocimiento, retroalimentación sobre padecimientos específicos para ayudar al autocuidado del paciente y convertir las terminologías médicas en indicaciones claras y entendibles para los pacientes.

1.4 Objetivos

Objetivo general

Desarrollar un servicio de web semántica a través de una base de conocimiento para compañeros digitales que integre ambos componentes de manera eficiente.

Objetivos específicos

1. Revisar los trabajos relacionados con bases de conocimientos en el área de la salud, así como su integración con servicios de web semántica para conocer las principales fortalezas y limitaciones de estas propuestas.

2. Desarrollar una base de conocimiento mediante el uso de mercado de datos (*Data Mart*), que extraiga la información de diferentes fuentes médicas, en particular de diabetes.

3. Construir el componente de web semántica, utilizando las técnicas de *Knowledge Graph* y *Scraping*, para consultas cuya respuestas son inexistentes en la base de conocimiento.

4. Realizar y probar la integración de la base de conocimiento y del componente de web semántica para detectar problemas y corregirlos.

5. Aplicar las pruebas de carga de trabajo utilizando el cuestionario NASA-TLX y las pruebas de percepción y usabilidad estética mediante el cuestionario *AttrakDiff* para evaluar el nivel de esfuerzo y experiencia de usuario en términos de atractivo,

funcionalidad y satisfacción de los usuarios durante la interacción con la solución propuesta.

1.5 Metodología

Para abordar eficazmente el objetivo del presente trabajo y garantizar la calidad de su ejecución, es esencial contar con una metodología sólida. La metodología desarrollada combina las buenas prácticas para el desarrollo de un proyecto y las actividades necesarias para llevar a cabo la implementación, de manera eficiente y precisa, la cual servirá de guía para todo el proceso.

Como se muestra en la Figura 1.2, la metodología a utilizar cuenta con cuatro fases: 1) Estudio preliminar, 2) Análisis y diseño, 3) Implementación, e 4) Integración y evaluación.

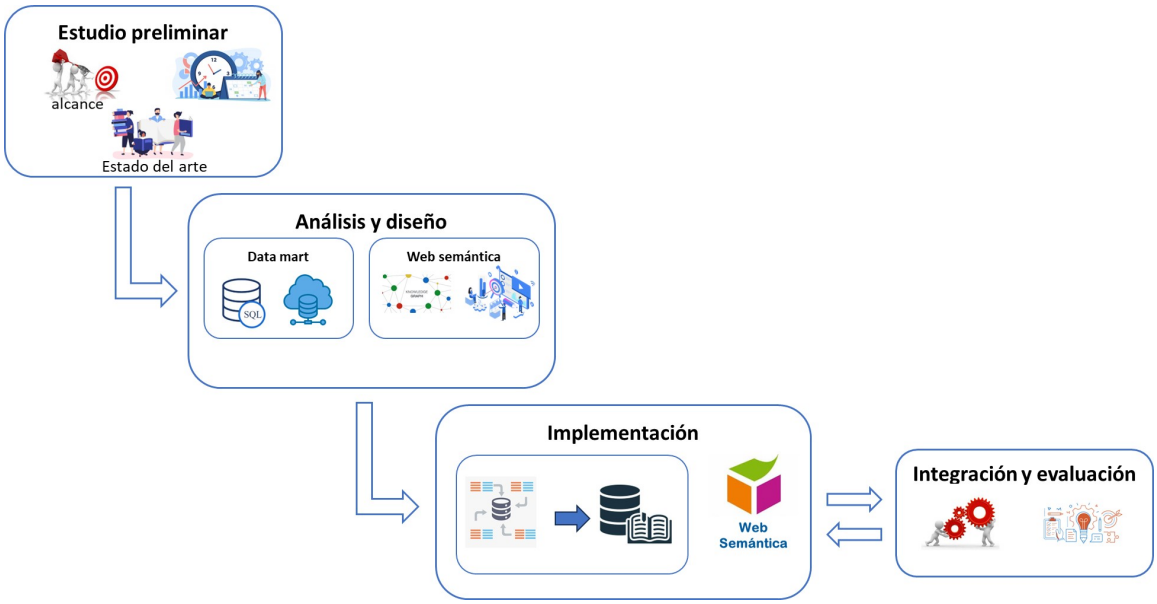


Figura 1.2: Metodología de desarrollo.

A continuación se describe cada una de las fases y sus actividades asociadas.

1. Estudio preliminar.
   - *a*) Definición del alcance del proyecto.
   - *b*) Elaboración del cronograma.
   - *c*) Estudio del estado del arte.
2. Análisis y diseño.
   - *a*) Identificación de las principales fuentes de datos para la recolección de la información del mercado de datos.
   - *b*) Definición del método a través del cual se realizarán las búsquedas del servicio de web semántica.

3. Implementación.

   a) Implementación del mercado de datos a partir de las fuentes identificadas.

   b) Conversión del mercado de datos en una base de conocimiento a través de técnicas como grafos de conocimiento y ontologías.

   c) Implementación del servicio de web semántica a través de la base de conocimiento.

4. Integración y evaluación.

   a) Integración de los componentes Base de conocimiento y Servicio de web semántica.

   b) Evaluación de la integración de ambas tecnologías.

La metodología planteada consiste en desarrollar, primeramente, una base de conocimiento con datos referentes a la salud, específicamente a la diabetes, como medicación, tratamiento, nutrición y ritmo de vida. Para ello, se implementará un mercado de datos, con información extraída de fuentes como archivos (Excel, CSV, XML, JSON) con información médica, servicios web a través del estándar REST y sistemas de almacenamiento en la nube. Luego, la información contenida en el mercado de datos, se convierte en conocimiento, mediante el enriquecimiento semántico, el cual consiste en la asignación de etiquetas semánticas, como clases, propiedades y sus relaciones. Además, el modelado de conocimiento, mediante ontologías y estructuras semánticas modeladas a través de grafos de conocimiento. A continuación, se desarrollará el servicio de web semántica, donde se integran tecnologías y estándares como RDF (*Resource Description Framework*), OWL (*Web Ontology Language*) y SPARQL, para permitir la búsqueda semántica y la recuperación de información. Finalmente, se integrarán ambas tecnologías y evaluación del proyecto resultante.

1.6 Propuesta de solución

Con la realización de este proyecto se pretende desarrollar un microservicio, utilizando la especificación de *RestFull Web Service*, que está enfocado en el desarrollo de un servicio de web semántica a través de una base de conocimiento. Esta integración consiste en que, una vez que se tiene la base de conocimiento, las consultas realizadas serán dirigidas primeramente a la base de conocimiento, que contendrá toda la información referente al tema tratado. Luego, de no existir la información solicitada, la búsqueda se realiza a través de la web semántica y, de esta manera, se enriquece la base de conocimiento. Este proceso tiene los siguientes beneficios:

- **Eficiencias en el acceso a datos**: la base de conocimiento previamente elaborada puede ser optimizada y estructurada para admitir consultas específicas y recuperar información, de manera eficiente, reduciendo los tiempos de respuesta en comparación con la realización de la búsqueda en tiempo real.

- **Reducción de latencia**: al tener una base de conocimiento local, se evita la latencia asociada con las consultas a la Web en tiempo real. Las consultas a través de la red pueden tener tiempos de respuesta variables, mientras que una base de conocimiento local generalmente ofrece acceso más rápido a la información.

- **Mejora de consistencia de datos**: la información contenida en la base de conocimiento puede ser curada, validada y mantenida para garantizar la consistencia de los datos. Esto es un aspecto crucial, ya que pueden existir consultas donde se requiere de información muy precisa.
- **Ahorro de ancho de banda**: evitar búsquedas directas en la Web puede resultar en un ahorro significativo en el ancho de banda, especialmente en situaciones donde la cantidad de datos transferidos entre el servicio y la web será considerable.
- **Mayor control sobre los datos**: tener una base de conocimiento garantiza un mayor control sobre la estructura y el formato de los datos, lo que facilita el diseño de consultas más específicas y adaptadas a las necesidades de los usuarios.
- **Privacidad y seguridad**: teniendo una base de conocimiento local, se tendrá control sobre la privacidad y seguridad de los datos, ya que en este proyecto se manejan datos sensibles y confidenciales.

Al mantener la búsqueda en una base de conocimiento local, se reduce el riesgo de exposición de información sensible a amenazas externas, teniendo en cuenta que dicha información está relacionada con temas de salud, donde la confidencialidad debe ser asegurada. Además, este tipo de búsquedas ofrece tiempos de respuesta más rápidos y una mayor disponibilidad de la información, ya que no depende de la conectividad a Internet, ni está sujeto a posibles interrupciones de la red. Esto puede ofrecer una experiencia de usuario más consistente y confiable. También se puede personalizar la información según las necesidades específicas del servicio, lo que mejora la relevancia y precisión de los resultados de búsqueda.

El resultado esperado es un servicio de web semántica que optimiza el acceso a datos, reduce la latencia, mejora la consistencia de la información y ofrece un mayor control sobre los datos. Además, al tener una base de conocimiento local, se garantiza la privacidad y seguridad de los datos, aspecto crucial al tratar información sensible y confidencial. Este proyecto aprovecha herramientas como *RestFull Web Service*, archivos planos, servicios web, RDF, OWL y SPARQL para lograr una integración eficaz entre la web semántica y las bases de conocimientos en el ámbito del cuidado de la salud.

1.7 Organización del documento

El presente trabajo está estructurado de la siguiente manera:

En el Capítulo 2 se detallan las principales definiciones asociadas a este trabajo de tesis, como datos, información, conocimiento, base de conocimiento, web semántica, entre otros. También se realiza un análisis de los trabajos relacionados con los componentes de Servicio de Web Semántica y Base de Conocimiento, así como con su integración.

En el Capítulo 3 se realiza el proceso de análisis y diseño de ambos componentes, enfatizando en la arquitectura, metodología a utilizar para el desarrollo del mercado de datos, así como la identificación de sus hechos y dimensiones, quedando conformado el modelo de datos. Además, se realiza el diseño de las transformaciones para la carga de los datos. Se identifican las clases, propiedades y relaciones que conforman la ontología propuesta. El diseño del Servicio de Web Semántica consiste en la definición de requerimientos, recursos y métodos a utilizar, así como la confección de su arquitectura.

El Capítulo 4 describe la implementación de los componentes de Base de Conocimiento, comenzando por el mercado de datos y el Servicio de Web Semántica. Se expondrán las herramientas utilizadas para la implementación y resultados obtenidos. En este punto se realizará el proceso de carga de los datos para poblar el mercado de datos y a su vez, la Base de Conocimiento resultante. Además, se explica el proceso de implementación del Servicio de Web Semántica desde la configuración del servidor y la Web *Scraping*. Asimismo, se añade a la implementación las consultas SPARQL como integración de ambos componentes, resultando el producto final.

En el Capítulo 5 se explica el proceso de realización de pruebas de la integración de ambos componentes. Se explica el Modelo V empleado para probar el mercado de datos. También se prueba la ontología realizada para validar su exactitud y coherencia. Además, se realizan pruebas de prototipo con usuarios finales, utilizando los cuestionarios NASA-TLX (*Task Load Index*) para medir la carga de trabajo y *AttrakDiff* para determinar la percepción y usabilidad estética.

Capítulo 2

Estado del arte

En este capítulo se exponen las principales definiciones y conceptos necesarios para comprender este trabajo de investigación (cf. Sección 2.1). Asimismo, se listan un conjunto de grafos de conocimiento de acceso abierto relacionados con la salud y de nivel general (cf. Sección 2.1.4). También se realiza el estudio de los trabajos relacionados, detallando sus características comunes, fortalezas y debilidades en relación con la solución propuesta (cf. Sección 2.2).

2.1 Marco teórico

En los últimos años ha habido un rápido desarrollo en el campo de los compañeros digitales, los cuales están diseñados para interactuar con los usuarios, a través de interfaces convencionales, que apoyan la realización de tareas como el cuidado de la salud. Este avance significativo ha remodelado intrínsecamente la forma en cómo se interactúa con la información y cómo se desarrolla el conocimiento a partir de ella.

En las siguientes subsecciones se definen varios conceptos importantes que son necesarios para tener un mejor entendimiento del enfoque que toma el presente trabajo.

2.1.1. Datos, información y conocimiento

Muchos son los autores que han definido los términos de datos, información y conocimiento, donde el tercero se deriva del segundo y este del primero, como se muestra en la Figura 2.1. Los datos son el componente semántico más pequeño y son la representación cruda de hechos, cifras o cualquier contenido que puede ser procesado por un sistema computacional. Son elementos básicos que por sí solos no tienen contexto ni significado propio.

La información, según Layzell Ward [56], es un conjunto de datos organizados y estructurados, de manera comprensible o con cierto significado, con el objetivo de comunicar un mensaje. Se refiere además a la contextualización de los datos, de forma que faciliten su análisis y la toma de decisiones. La información tiene significado, es decir, relevancia y propósito.

Por otra parte, el conocimiento es el proceso de comprensión de dicha información que se adquiere a través de su análisis, experiencia o aprendizaje [16]. Aunque existen criterios de igualdad entre datos, información y conocimiento, la diferencia radica en el nivel de comprensión y procesamiento, donde la información puede considerarse como datos procesados, en tanto que el conocimiento implica un nivel más alto de comprensión.

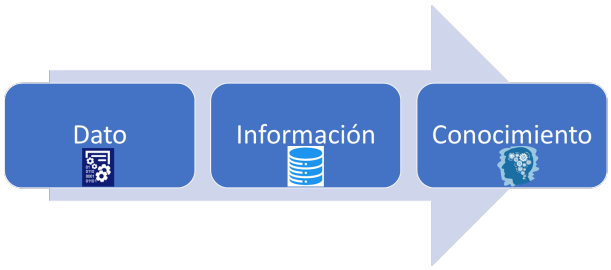


Figura 2.1: Relación entre datos, información y conocimiento.

2.1.2. Base de conocimiento

Una base de conocimiento, según Wu [58], es un conjunto de sistemas de conocimientos que incluyen los conceptos de organización del conocimiento y la tecnología de la información para la recolección, organización, formalización y estandarización del conocimiento de manera eficiente. Se refiere, además, a una colección organizada y bien estructurada de la información o datos que se pueden utilizar para ayudar a la toma de decisiones y a la solución de problemas. Almacenan información fáctica o basada en los hechos en forma de relaciones entre entidades.

Las características de una base de conocimiento son [15]:

- **Representación estructurada del conocimiento**: la información en una base de conocimiento se organiza de manera estructurada, utilizando grafos o relaciones entre entidades, conceptos y hechos. Para ello, se utilizan ontologías y esquemas de datos que definen cómo se relacionan los distintos elementos dentro de la base de conocimiento.

- **Uso de ontologías**: en una base de conocimiento, las ontologías son fundamentales para definir categorías, propiedades y relaciones entre los conceptos, lo que ayuda a organizar el conocimiento de una manera coherente y comprensible para las computadoras.

- **Consultas semánticas**: una base de conocimiento permite realizar consultas semánticas, que no sólo buscan datos, sino también información relacionada que esté basada en el significado. Este proceso se realiza a través de lenguajes como SPARQL que interpretan el contexto y la intención detrás de las consultas.

- **Interoperabilidad**: las bases de conocimiento suelen ser interoperables. Esta característica significa que pueden integrarse con otras bases de datos o sistemas a través de estándares abiertos como RDF y OWL.

- **Automatización y autogestión**: la mayoría de las bases de conocimiento incluyen capacidades de automatización, lo que les permite actualizarse y expandirse automáticamente a medida que se integre nueva información o se generen nuevos datos.

- **Escalabilidad**: las bases de conocimiento pueden manejar grandes volúmenes de datos y seguir siendo eficientes a medida que el conocimiento crece. Esta capacidad es fundamental en aplicaciones que requieran almacenar y procesar grandes cantidades de información, como investigaciones científicas o minería de datos.

2.1.3. Web tradicional

La Web tradicional o Web 1.0 es la primera generación de la *World Wide Web* (WWW), la cual se caracteriza por representar la información de manera estática y con limitada interactividad. Los sitios web consisten en documentos estáticos compuestos por textos e imágenes, codificados en HTML (*Hypertext Markup Language*). Estos documentos también proporcionan información que los usuarios pueden leer, pero sin la posibilidad de interactuar con el contenido dinámicamente [11]. En la Web tradicional, la interacción con el usuario es limitada, así como la personalización de la experiencia del usuario, por lo que la Web se percibe más como un conjunto estático de documentos en línea que como una plataforma dinámica e interactiva.

La Web 1.0 sentó las bases para el posterior desarrollo de tecnologías más avanzadas, tales como la Web 2.0 y la Web semántica, que incorporan características de interactividad y representación semántica de los datos. La Web semántica busca ir más allá de una simple representación visual de la información, por lo que se centra en representar los datos de manera que las computadoras puedan entender y comprender su significado y contexto, facilitando una búsqueda y recuperación de información más eficiente e inteligente.

2.1.4. Web semántica y grafo de conocimiento

La Web semántica es una extensión de la WWW, que se basa en proporcionar datos bien definidos, enlazados y procesables por computadora, de modo que garanticen y brinden información relevante al usuario. Además, la Web semántica posee el potencial de analizar y sintetizar información de manera eficaz, proporcionando respuestas pertinentes a consultas complejas. Dicha información debe ser almacenada en una estructura de datos, de manera que sea más fácil de buscar para generar conocimiento. Estas estructuras pueden ser sintetizadas a través de grafos de conocimiento (*Knowledge Graph*), los cuales son una representación de objetos, conceptos, situaciones o eventos y las relaciones que existen entre ellos. Son muy utilizados para problemas que requieran análisis complejos, razonamientos lógicos, etc. Google, por ejemplo, utiliza grafos de conocimiento como herramienta central para almacenar información que luego facilita como respuestas inmediatas, precisas y detalladas a consultas sobre un tema en específico, sin que el usuario tenga que hacer búsquedas secundarias. Estas respuestas contienen variedad de información y formatos, ventaja que brindan los grafos de conocimiento [40].

A menudo, los grafos de conocimiento suelen confundirse con ontologías, a pesar de que hay diferencias sustanciales entre ambos conceptos. Studer et al., [50] definen una ontología como una especificación explícita de conceptos, interpretados por una computadora de forma simplificada y abstracta. Las ontologías son de utilidad para representar formalmente las entidades en un grafo, basándose en múltiples taxonomías.

Varios son los grafos de conocimiento disponibles de acceso abierto (*open source*) y muy reconocidos como [40]:

- **DBpedia**: es una base de conocimiento que contiene datos provenientes de varias fuentes para extraer contenido ya estructurado de la información presente en proyectos de Wikimedia. Ofrece una forma estructurada y semántica de acceder a la información.
- **YAGO**: es una base de conocimiento semántico que, combinando información de varias fuentes, crea una gran ontología con múltiples relaciones y entidades.

- **Freebase**: es una base de datos abiertamente compartida con gran cantidad y diversidad de información y conocimiento en general.
- **Wikidate**: es una base de datos en diversos idiomas, que brinda soporte a proyectos de Wikimedia a través de la recopilación de datos estructurados.
- **OpenCyc**: considerada como la base de conocimiento más completa del mundo, presenta motores de razonamiento de conocimiento general basado en el sentido común.

Además de las bases de conocimiento mencionadas, existen otras más específicas para el área de salud, como son:

- **Healthcare Graph**: es un grafo de conocimiento que centraliza datos de salud, específicamente detalles específicos de padecimientos como síntomas, tratamientos, interacciones medicamentosas, etc.
- **BioPortal**: contiene un conjunto de ontologías que facilitan la construcción de grafos de conocimiento en el ámbito biomédico, incluyendo aspectos relacionados con la diabetes.
- **Disease Ontology**: es una ontología de enfermedades específicas, que ofrece una jerarquía estructurada de enfermedades humanas, incluyendo diabetes, sus tipos y subtipos.

2.1.5. Servicio web

Un servicio web, según IBM [29], está conformado por un grupo de aplicaciones relacionadas entre sí, que pueden ser invocadas a través de Internet. De esta forma, se definen como aplicaciones modulares, autónomas y autodescriptivas, con capacidad para publicar, localizar e invocar en la web. Un servicio web se basa en el intercambio de mensajes con otra aplicación, permitiendo el envío y la recepción de información.

Algunas características de los servicios web son las aiguientes [29]:

- **Modularidad**: se pueden combinar servicios web sencillos para crear servicios web de mayor complejidad, mediante técnicas de flujo de trabajo o a través de la invocación de servicios web de niveles inferiores desde el servicio superior.
- **Autonomía**: los servicios web funcionan de manera independiente, i.e., no requieren software adicional, sino que sólo dependen de un lenguaje de programación que soporte XML (*Extensible Markup Language*) y HTTP (*Hypertext Transfer Protocol*). En el servidor se necesita un servidor web y un motor de *servlets*. El cliente y el servidor pueden estar en entornos diferentes. Además, los servicios web pueden activar una aplicación existente sin la necesidad de escribir código nuevo.
- **Autodescripción**: los servicios web se describen por sí mismos. El cliente y el servidor sólo necesitan entender el formato y el contenido de los mensajes de solicitud y respuesta. La definición del formato de dicho mensaje va incluido dentro del mismo mensaje, de esta manera se elimina la necesidad de repositorios con metadatos externos y herramientas de generación de código.

- **Independencia**: los servicios web son independientes de la plataforma. Están sustentados en un conjunto específico de estándares abiertos basados en XML, diseñados para garantizar la interoperabilidad entre servicios web y clientes en una variedad de sistemas y lenguajes de programación.

2.1.6. Mercado de datos

Por otra parte, el presente trabajo pretende diseñar e implementar una base de conocimiento mediante el uso de un mercado de datos, el cual es una estructura especializada y diseñada para recopilar, organizar y proporcionar acceso a datos específicos de un área de negocio, permitiendo así una gestión más eficiente y un análisis detallado de la información relacionada con ese dominio en particular. Un mercado de datos es un conjunto de hechos y datos organizados para soporte decisional, que se basa en la necesidad de un área o departamento específico, teniendo sólo sentido para el personal de ese departamento y sus datos no tienen por qué tener las mismas fuentes que los de otro mercado de datos [57]. Son un subconjunto de igual implementación a los almacenes de datos (*Data Warehouse*), por lo que adquieren las siguientes características [30]:

- **Orientado a temas**: los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo objeto o evento del mundo real queden unidos entre sí.
- **Variable en el tiempo**: los cambios que se produzcan en los datos, a lo largo del tiempo, quedan registrados de manera que los informes que se generen deben reflejar dichas variaciones.
- **No volátil**: la información almacenada no se puede modificar ni eliminar, por lo que se convierte en información de sólo lectura.
- **Integrado**: los datos que integran todos los sistemas operacionales de la organización o el área de negocio, se mantienen en las mismas bases de datos de manera consistente.

2.2 Trabajos relacionados

Como parte del estudio y análisis del estado del arte sobre trabajos relacionados con bases de conocimientos y servicios de web semántica, se definieron una serie de preguntas específicas para cada componente, con el fin de determinar semejanzas y diferencias con el presente trabajo, las cuales se describen a continuación.

2.2.1. Preguntas para la Web semántica

Con el fin de identificar analogías y contrastes entre el presente trabajo y otros existentes en cuanto al desarrollo del Servicio de Web Semántica, se proponen las siguientes preguntas que permitirán explorar aspectos fundamentales relacionados con dicho servicio:

- P1- ¿Qué técnicas de procesamiento de información se utilizan?
- P2- ¿Qué técnicas de recuperación de información se emplean?

- P3- ¿A qué dominio o aplicaciones específicas va dirigido el servicio?

A continuación, se describen algunos de los trabajos relacionados más relevantes referentes a los servicios de Web semántica:

Bansal et al. [5] proponen un servicio de Web semántica en el que, mediante una descripción semántica de los servicios, el motor de búsqueda implementado produce resultados de manera óptima, encontrando cualquier composición condicional, ya sea secuencial o no secuencial ante consultas determinadas. Además, genera de forma automática una descripción OWL-S del servicio compuesto, la cual es utilizada en la fase de ejecución para que las posteriores búsquedas de este servicio tengan una coincidencia directa. El servicio utiliza técnicas de inferencia semántica para el procesamiento de las consultas realizadas por el usuario. Aunque se basa en grafos acíclicos, dirigidos y condicionales para la recuperación de información, presenta un dominio de aplicación de propósito general.

Zheng et al. [61] describen la herramienta CDSMKB (*Chronic Disease Self-Management Knowledge Base*), la cual utiliza OWL y marco web semántico de Jena para transformar los datos de manera semántica y realizar inferencias sobre la información ya presente en un conjunto de datos RDF. Incluyen la implementación de un CDSMS (*Mobile Chronic Disease Self-Management System*) basado en la integración del conocimiento del dominio en CDSMKB y los datos semánticos del historial clínico del paciente. Este sistema utiliza técnicas de inferencia semántica para el procesamiento de la información, la cual se obtiene a través de reglas y consultas SPARQL previamente definidas. Aunque su dominio es en el área de salud y utiliza ontologías para la representación del conocimiento y así definir y compartir conceptos y relaciones, está diseñada como aplicación móvil.

Gulzar y Ahmed [25] realizan un estudio de las principales tecnologías de la Web semántica para la búsqueda y el manejo de enfermedades no transmisibles (ENT), como la diabetes. Se tienen en cuenta las principales áreas de aplicación de la Web semántica en atención médica, como bases de conocimiento de enfermedades crónicas. Para ello, en este trabajo se propone un sistema para la predicción y medicación de las ENT. Como primer paso se realiza la construcción de una base de conocimiento a partir del historial clínico del paciente, registros médicos, etc. Luego se detectan las enfermedades a través de un mecanismo de inferencia como SWRL (*Semantic Web Rules Language*).

El trabajo de Fayçal y Abdelkamel [17] se basa en la implementación de un servicio web semántico enfocado en la industria farmacéutica. Para ello se realiza la construcción de dos ontologías, una para el dominio farmacéutico (ontología de dominio farmacéutico) con conceptos, instancias, atributos, axiomas, normas y restricciones en el campo médico y otra de servicios web farmacéuticos, como agentes de software, aplicaciones sanitarias, computación en la nube, etc. Ambas tienen el objetivo de garantizar la calidad en los servicios médicos. A pesar de estar enfocado al área de la salud, su utilización se centra más hacia especialistas de esta rama, como médicos, enfermeros, farmacéuticos, y no va enfocado al paciente en cuestión. Además, su aplicación es general en cuanto a temas de salud.

Wang et al. [55] proponen el diseño e implementación de un grafo de conocimiento con información médica para un sistema PHR (*Personal Health Records*), proporcionando métodos de transformación semántica de los registros médicos en bases de datos relacionales a datos semánticos. Esta propuesta aplica OWL y reglas Jena para la representación de los datos médicos en el grafo de conocimiento. Wang et al. proponen, además, una arquitectura de integración semántica entre el sistema PHR y los sistemas de información hospitalarios como farmacia, laboratorios, radiología, etc., los cuales son sistemas heterogéneos con diferentes tipos de bases de datos. El grafo de conocimiento, que se diseñó

en la propuesta, está centrado en el paciente y se utiliza la técnica de minería de datos semánticos para el descubrimiento de nuevos patrones de conocimiento y correlaciones entre ellos.

2.2.2. Preguntas para la base de conocimiento

- P4- ¿Qué SGBD [1] se utiliza para la gestión de los datos?
- P5- ¿Cómo se realiza el proceso de alimentación de la base de conocimiento para cargar y actualizar los datos? (nunca, fija, manual, automática, en tiempo real).
- P6- ¿Qué tipo de acceso tiene la base de conocimiento?

Rippa y Kasatkina [49] plantean el diseño de herramientas de base de conocimiento y una arquitectura de desarrollo en el entorno de la Web semántica. Además, proporcionan una breve descripción de la plataforma de software utilizada para operar las bases de conocimiento, que pueden actuar como proyectos de realidad virtual en el contexto de la Web semántica. En este sentido, la creación de una base de conocimiento supone una secuencia de etapas, que comienza desde la formulación de los objetivos hasta el modelado de datos y la construcción de conocimiento.

Yin et al. [60] realizan el diseño e implementación de un grafo de conocimiento sobre diabetes, a través de registros médicos electrónicos para el hospital Ruijing de Shanghai. Este proyecto realiza la finalización de una base de conocimiento, que consiste en añadir nuevos hechos o entidades a una base de conocimiento ya existente, por lo que pueden surgir nuevas relaciones entre las entidades. Para ello, Yin et al. realizan un análisis de los datos originales con el objetivo de eliminar o reducir los datos ruidosos, a través de algoritmos tradicionales de agrupamiento como K-Means. Luego, como segundo paso, construyen un grafo de conocimiento sobre diabetes, en el que identifican y definen los conceptos de entidades y sus relaciones.

CoreKB [22] es una herramienta o plataforma web para la búsqueda a través de una base de conocimiento de agrupaciones entre expresiones genéticas y cáncer, extraídos de la literatura científica de PubMed. Esta herramienta utiliza lenguaje natural y facetas estructuradas, lo que facilita la búsqueda por hechos y entidades. Además, permite a los médicos e investigadores, la búsqueda de hallazgos médicos confiables y valiosos. Este proyecto integra web semántica y base de conocimiento y utiliza técnicas de Procesamiento de Lenguaje Natural (PLN) para el procesamiento de las consultas de búsqueda, mediante grafos de conocimientos, con temas referentes a la salud, pero específicamente del cáncer.

OC-2-KB (*Obesity and Cancer to Knowledge Base*) [38] es un sistema que crea, de manera automática, una base de conocimiento de la Web semántica a partir de resúmenes de PubMed. Se divide en dos módulos o procesos fundamentales; el primero, *offline*, donde se crean los diccionarios de entidades y predicados candidatos que son relevantes para el dominio; se crea una vez, como parte de la configuración del sistema; el segundo proceso es *online*, en el que se extraen los datos de la literatura biomédica para construir la base de conocimiento de la Web semántica. Aunque se manifiesta una relación entre Web semántica y la base de conocimiento, esta última se basa solamente en pacientes que padecen de cáncer y obesidad.

[1] SGBD: Sistema Gestor de Bases de Datos

2.2.3. Comparativa de trabajos

La Tabla 2.1 presenta los diferentes trabajos, descritos en Secciones 2.2.1 y 2.2.2, los cuales han sido clasificados según las técnicas para el procesamiento y la recuperación de la información, el dominio de aplicación, el sistema de gestión de las bases de datos (SGBD), el proceso de alimentación de los datos y el tipo de acceso.

Tabla 2.1: Tabla comparativa del estado del arte

Trabajos analizados	Técnica de procesamiento	Técnica de recuperación	Dominio de aplicación	SGBD	Proceso de alimentación	Tipo de acceso
Bansal et al.	Inferencia semántica	Razonamiento ontológico y Consultas SPARQL	General	No aplica	No aplica	No aplica
Zheng et al.	Inferencia semántica	Consultas SPARQL	Salud	No aplica	No aplica	No aplica
Gulzar y Ahmed	Inferencia semántica	Consultas SPARQL	Salud (ENT)	No aplica	No aplica	No aplica
Fayçal y Abdelkamel	PLN	Razonamiento ontológico	Salud (Especialistas)	No aplica	No aplica	No aplica
Wang et al.	Minería de datos semánticos	Consultas SPARQL	Salud	No aplica	No aplica	No aplica
Rippa y Kasatkina	No aplica	No aplica	No aplica	No se especifica	Fija	No se especifica
Yin et al.	No aplica	No aplica	No aplica	Registros médicos electrónicos	Automátca	No se especifica
CoreKB	PLN	Grafos de conocimiento	Salud (Cáncer)	PostgreSQL	No se especifica	No se especifica
OC-2-KB	PLN	Consultas de SPARQL	Salud (Cáncer-Obesidad)	GraphDB [2]	No se especifica	No se especifica
Trabajo a desarrollar	*Data Mart* (ETL)	Consultas SPARQL *Scraping*	Salud (Diabetes)	PostgreSQL	Automática	A nivel de accesos

Técnicas de procesamiento de la información

La mayoría de los trabajos emplean técnicas de inferencia semántica, PLN y minería de datos semánticos para procesar la información, a través de reglas lógicas y ontologías, análisis e interpretación del lenguaje humano y descubrimiento de patrones y correlaciones, respectivamente. Sin embargo, el trabajo desarrollado en esta tesis propone la utilización de un mercado de datos, a través de ETL para recuperar la información. Un mercado de datos está diseñado para almacenar subconjuntos de datos específicos, lo que facilita consultas más rápidas y eficientes al centrarse en un área de temática. La utilización de ETL permite la integración de múltiples fuentes y garantiza que el mercado de datos se mantenga actualizado. Además, a medida que crece el volumen de datos semánticos, un mercado de datos bien diseñado puede escalar fácilmente, adaptándose a las necesidades de recuperación de información.

[2]GraphDB es un sistema de gestión de bases de datos de grafos. Una base de datos de grafos es un tipo de base de datos que utiliza estructuras de datos de grafo para representar y almacenar información.

Técnicas de recuperación de la información

La recuperación de información se realiza mayormente a través de consultas SPARQL, debido a su capacidad para interactuar eficientemente con datos estructurados en formato RDF. El trabajo desarrollado en esta tesis también propone la utilización de consultas SPARQL para la recuperación de la información que está almacena en la base de conocimiento y *scraping* para recuperar información de la Web.

Dominio de aplicación

El dominio de aplicación en la mayoría de los trabajos analizados está relacionado con el campo de la salud, cubriendo ENT, cáncer y especialidades médicas. En particular, el trabajo desarrollado en esta tesis está enfocado a pacientes con diabetes. En este sentido, maneja información sobre la medicación, la alimentación, el diagnóstico y el tratamiento de la diabetes.

SGBD, proceso de alimentación y tipo de acceso a los datos

En la mayoría de los casos, aunque no se especifica o no aplica el uso de SGBD, así como tampoco se evidencia el proceso de alimentación de los datos ni el tipo de acceso a ellos, PostgreSQL se presenta en dos de ellos. El trabajo propuesto en esta tesis está desarrollado bajo el SGBD de PostgreSQL. El proceso de alimentación de la base de datos se hace de manera automática, ya que una vez que se obtiene la información mediante *scraping*, esta es almacenada en la base de conocimiento. Asimismo, el tipo de acceso es a través de niveles de acceso, especificando el nivel de administrador y de paciente.

En resumen, el trabajo expuesto en el presente documento resalta el uso de un mercado de datos mediante procesos de ETL para el procesamiento de datos relacionados con la diabetes. Integra consultas SPARQL y un sistema de *scraping* automatizado, que permite, de manera continua, alimentar la base de datos de PostgreSQL con información relevante.

Capítulo 3

Análisis y diseño del sistema

En este capítulo se realiza el proceso de análisis y diseño de la base de conocimiento y del servicio de web semántica. Se define el proceso de desarrollo de la base de conocimiento, desde el diseño del mercado de datos hasta la fase de concepción de una ontología semántica, enfatizando en la arquitectura, la metodología a utilizar para el desarrollo del mercado de datos, así como la identificación de sus hechos y dimensiones. De esta manera queda conformado el modelo de datos y el diseño de las transformaciones para la carga de las tablas de hechos y dimensiones. Además, se diseña la base de conocimiento a través de la definición de clases, subclases, relaciones e instancias de la ontología (cf. Sección 3.1). Luego se detalla el diseño del servicio de web semántica, en el que se define su arquitectura y el proceso de búsqueda de información relacionada con la diabetes (cf. Sección 3.2).

3.1 Base de conocimiento

El proceso de desarrollo de la base de conocimiento consiste en la realización de un mercado de datos para su posterior conversión a una base de conocimiento. Este proceso implica transformar los datos estructurados en información semántica que pueda ser interpretada y utilizada por sistemas de inteligencia artificial para el razonamiento y el apoyo a la toma de decisiones. Para ello se tiene en cuenta los siguientes pasos:

1. **Planificación de la semántica**: implica la comprensión de los requisitos del negocio y de cómo se utilizará la información en el proyecto, así como la identificación de conceptos o entidades del negocio.

2. **Selección de herramientas apropiadas**: requiere el análisis y la selección de herramientas y tecnologías que faciliten la captura y representación de la información, ya sea bases de datos, motores de inferencia, herramientas de consultas semánticas, etc.

3. **Desarrollo de ontologías y modelado semántico**: implica el desarrollo de una ontología que represente los conceptos definidos, sus propiedades y relaciones en el dominio de los datos.

4. **Diseño del esquema del mercado de datos**: requiere la definición de hechos, dimensiones y medidas, las cuales deben coincidir con los conceptos definidos en la ontología.

5. **Proceso de ETL (*Extraction, Transformation and Loading*)**: implica el proceso de extracción, transformación y carga de los datos, así como la captura de los metadatos semánticos.

6. **Enriquecimiento semántico**: supone el enriquecimiento con metadatos adicionales que proporcionen contexto y significado, tales como adición de etiquetas semánticas, descripciones detalladas y relaciones adicionales entre entidades.

7. **Integración de herramientas de análisis semántico e inteligencia artificial**: incluye la integración de herramientas y sistemas de análisis semántico y de inteligencia artificial con el mercado de datos. También supone la utilización de motores de inferencia y herramientas de consulta semántica para realizar análisis avanzados de los datos.

8. **Evaluación y refinamiento continuos**: consiste en la evaluación de la integración de la semántica en el mercado de datos.

3.1.1. Diseño del mercado de datos

Existen varias metodologías que rigen el proceso de diseño e implementación del un mercado de datos como Inmon, HEFESTO y Kimball:

- **Inmon**: es una metodología relacional, propuesta por Inmon [30], que plantea la construcción de un mercado de datos por separado para cada área o proceso de negocio. Los datos se ingresan al almacén de datos (*Data Warehouse*) de forma integrada, por lo que actúa como única fuente de datos para varios mercados de datos. Esta característica hace que el proceso de ETL sea más sencillo y menos propenso a fallas, pero se requiere una operación adicional en dicho proceso, ya que los mercados de datos se crean después de haberse generado el almacén de datos. A este enfoque se le denomina descendente (*Top-Down*).

- **HEFESTO**: esta metodología propone un enfoque con base en la construcción robusta y meticulosa de los almacenes de datos, enfatizando en la calidad y la precisión en el desarrollo de los mismos [6]. Esta característica de ser un enfoque detallado y exhaustivo hace que se tenga mayor rigor y control en la documentación, así como una mayor resistencia a cambios, dificultando la adopción de enfoques más flexibles y ágiles.

- **Kimball**: la propuesta de Kimball [32] es una metodología relacional. Utiliza un enfoque ascendente (*Bottom-Up*) que propone primero la implementación de los mercados de datos y luego la integración de estos en el almacén de datos. No necesita tener un modelo completamente normalizado, lo que garantiza rapidez en el proceso inicial de almacenamiento de datos. Esta metodología, aunque tiene menor flexibilidad de modificación, presenta una estructura bien definida y optimizada para lograr la eficiencia y el rendimiento.

Teniendo en cuenta estas metodologías y haciendo un análisis de sus pros y contras, para la construcción del mercado de datos, fue seleccionada la metodología planteada por Kimball, ya que es ideal para proyectos que requieren un desarrollo rápido y eficiente. Tomando como base el enfoque ascendente, que esta presenta, fue adaptada a las características del proyecto, haciendo énfasis en el proceso de ETL de los datos. Esto simplifica la etapa inicial al no requerir una integración total de los datos del almacén completamente normalizado y reduce el riesgo a fallos en las primeras etapas del proyecto. Además, este enfoque facilita la integración modular y específica, permitiendo la obtención de beneficios

de manera rápida, lo cual es ventajoso en el ámbito de la salud, donde la rapidez y la capacidad de adaptación son cruciales.

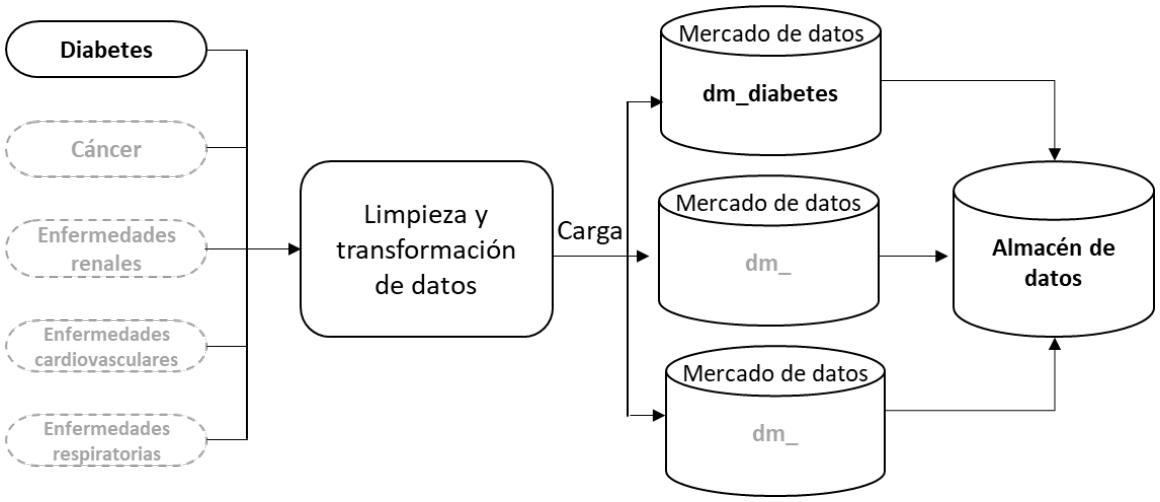


Figura 3.1: Enfoque de desarrollo basado en la metodología de Kimball.

La Figura 3.1 muestra el enfoque de desarrollo del mercado de datos `dm_diabetes` de manera dimensional. También incluye la definición de otras posibles áreas de salud o ENT (Enfermedades no Transmisibles) que pueden ser integradas como mercado de datos, en caso de requerir una extensión del trabajo de investigación, siguiendo el mismo flujo de análisis de información.

Para facilitar el proceso de toma de decisiones en el mercado de datos, se requiere la identificación de las necesidades de información asociadas a las posibles preguntas que haría un paciente sobre un determinado tema de diabetes, ya que estas constituyen la base para el diseño del mercado de datos. La información se clasifica en tres grupos o áreas de análisis en función del proceso de negocio, para este caso la diabetes: `Diagnóstico`, `Tratamiento` y `Alimentación`, las cuales se definen como tablas de hechos del sistema.

Las Tablas 3.1 y 3.2 muestran la descripción de los hechos y las dimensiones identificadas, respectivamente.

Tabla 3.1: Hechos del mercado de datos

Hechos	Descripción
`hech_tratamiento`	Contiene la información relacionada con el tratamiento asociado a un paciente de diabetes, teniendo en cuenta el tipo de diabetes y el medicamento adecuado.
`hech_alimentacion`	Contiene la información concerniente a la alimentación y a los nutrientes que debe consumir un paciente de diabetes.
`hech_diagnostico`	Contiene la información relacionada con el diagnóstico asociado a un paciente de diabetes, teniendo en cuenta el tipo de diabetes y las pruebas realizadas.

Tabla 3.2: Dimensiones del mercado de datos

Dimensiones	Descripción
dim_paciente	Recoge los datos relacionados con el paciente, así como su peso, talla, género, etc.
dim_medicamentos	Almacena los medicamentos asociados a la diabetes.
dim_sintomas_prediabetes	Guarda los síntomas que puede tener un paciente antes de que se le sea detectada la enfermedad.
dim_sintomas_post	Reúne los síntomas que puede tener un paciente durante la enfermedad.
dim_otras_enfermedades	Agrupa las posibles enfermedades extras que pueden estar asociadas a un paciente de diabetes.
dim_glucemia	Genera los niveles de glucemia que puede tener un paciente de diabetes.
dim_alimentos	Contiene los tipos de alimentos y nutrientes que puede consumir un paciente de diabetes.
dim_tipo_diabetes	Describe los tipos de diabetes.
dim_test_diagnostico	Almacena los tipos de pruebas de diagnóstico que se realizan a un paciente de diabetes.

Modelo de datos del mercado de datos

El modelo de datos representa la relación entre las tablas de hechos y dimensiones identificadas. De esta manera, los datos del negocio quedan reflejados en forma de cubos de datos. El modelo propuesto para la solución del mercado de datos presenta una topología de Constelación de Hechos, ya que existen dimensiones compartidas para más de una tabla de hechos. La Figura 3.2 muestra dicha representación, así como las relaciones entre las tablas identificadas.

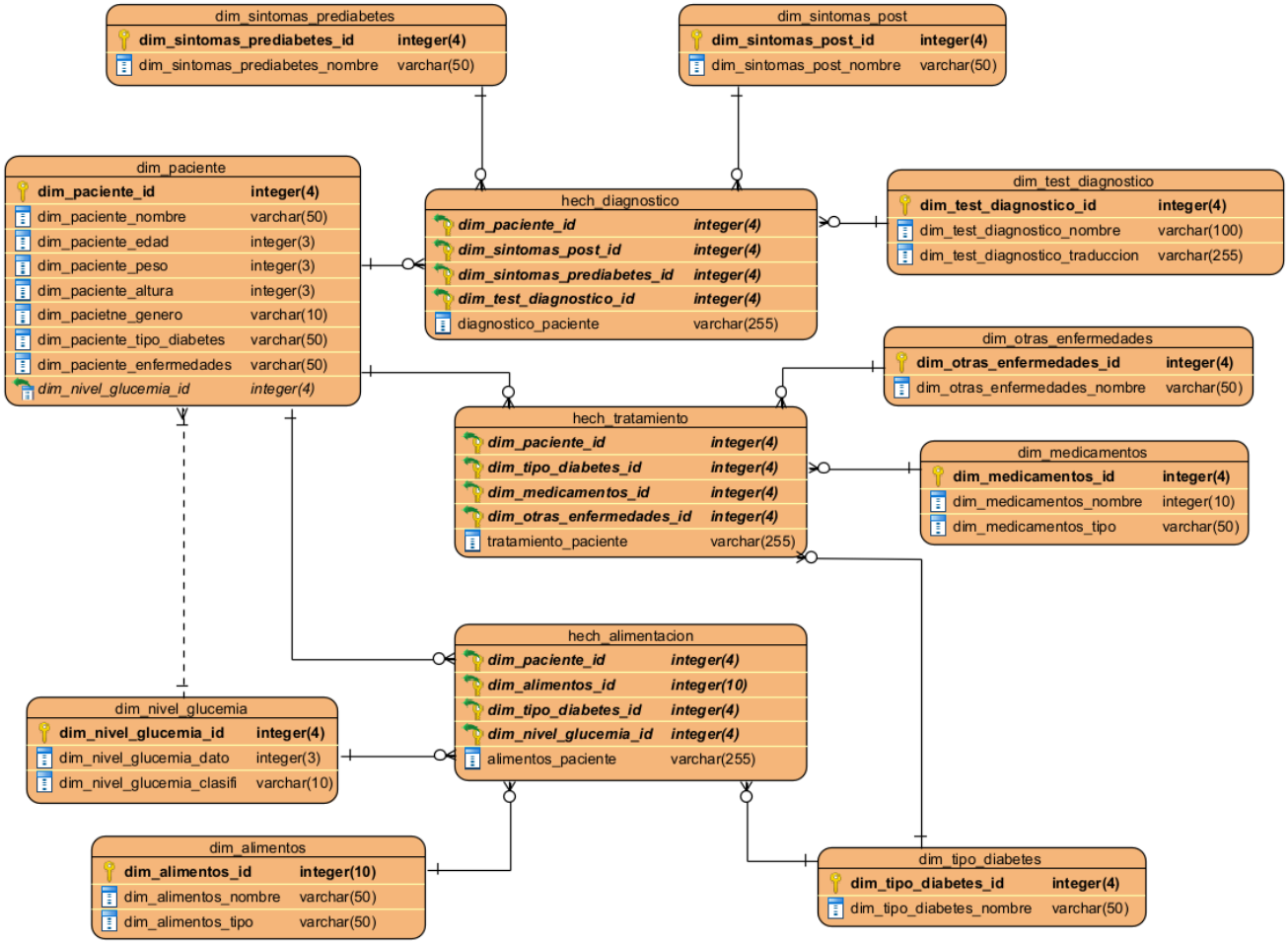


Figura 3.2: Modelado de datos de mercado de datos.

Existe una gran variedad de herramientas que apoyan el proceso de análisis y diseño de *software*. Para el modelado de los datos del mercado de datos se utilizó Visual Paradigm for UML 17.1, la cual es una herramienta CASE (*Computer Aided Software Engineering* o Ingeniería de Software Asistida por Computadora) que permite el modelado de todo tipo de diagramas, es compatible entre ediciones y genera un *script* para el SGBD (Sistema Gestor de Bases de Datos) de PostgreSQL.

Proceso de integración de datos del mercado de datos

En el proceso de integración, se realiza el diseño de las transformaciones como parte del proceso de ETL. Para ello, se utilizan diferentes estrategias de integración de datos, según Basallo et al. [4]:

- **Replicación de datos**: implica crear y mantener copias de la base de datos donde, generalmente, un servidor contiene la copia primaria de las bases de datos y otros mantienen las copias esclavas de las mismas.
- **Integración de información empresarial**: consiste en la creación de un intermediario que realice la función de canal de consulta y representación de la información recuperada y contenga los directorios de estas bases de datos.
- **Integración de aplicaciones empresariales**: se trata de integrar varias aplica-

ciones con tecnologías no compatibles, permitiendo que estas aplicaciones se comuniquen entre sí e intercambien información.

- **Proceso de ETL**: tiene por objetivo extraer de datos de sistemas fuentes, realizar el proceso de transformación y cargarlos a un sistema destino.

Esta última estrategia será utilizada para el proceso de ETL de la solución, ya que se prevé la extracción de información de fuentes de datos médicos, para luego transformarlos según las necesidades de los pacientes y ajustar los diferentes conceptos médicos a terminologías entendibles por el paciente.

Diseño de las transformaciones

Carga de las dimensiones: primeramente, se extraen los datos de fuentes como archivos en formato .json, obtenidos durante el proceso de *scraping* al sitio web BioPortal[3] a través de su API y/o, en algunos casos, se introducen directamente o se generan de manera automática. Se seleccionan los datos correspondientes a las dimensiones a cargar, para luego realizar el proceso de limpieza y transformación e insertarlos en la base de datos `dm_diabetes`. La Figura 3.3 muestra el diseño de las transformaciones para la carga de las dimensiones.

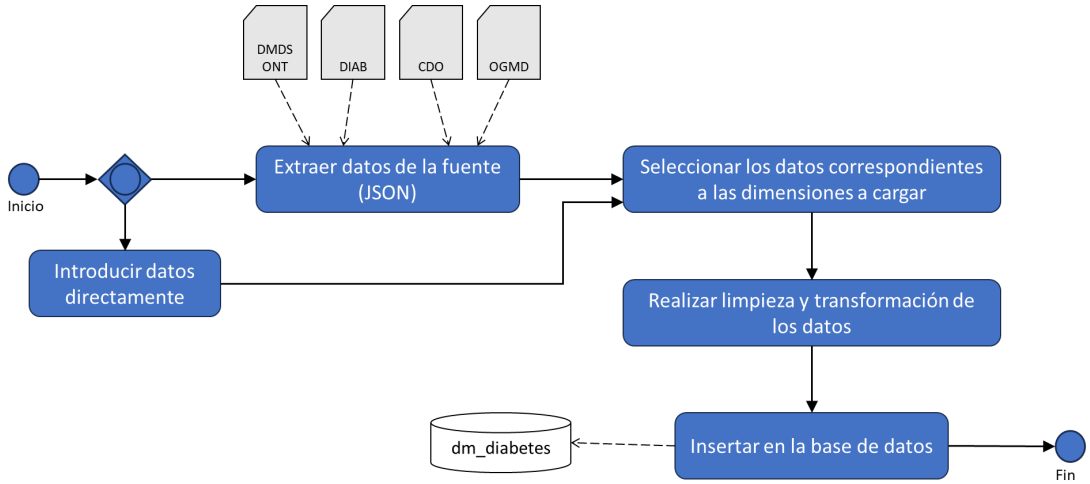


Figura 3.3: Diseño de las transformaciones para la carga de dimensiones.

Carga de hechos: luego de haber cargado las tablas de dimensiones, se obtiene y se inserta la información CDC (*Change Data Capture*), se extraen los datos de las fuentes identificadas, y se realiza la limpieza y transformación. A continuación, se buscan las llaves primarias de cada una de las dimensiones para obtener el nombre del campo, tanto en la fuente como en la base de datos destino, para luego ser validados de manera que cumplan con las restricciones definidas. Si las llaves no coinciden con las de la base de datos destino, se almacenan en archivos Excel para su posterior análisis. Si coinciden dichas llaves, se obtiene la información requerida sobre los tratamientos o alimentación de un paciente y se inserta dicha información en la base de datos destino `dm_diabetes`. La Figura 3.4 muestra el diseño de las transformaciones para la carga de hechos.

[3]BioPortal: Repositorio de ontologías biomédicas

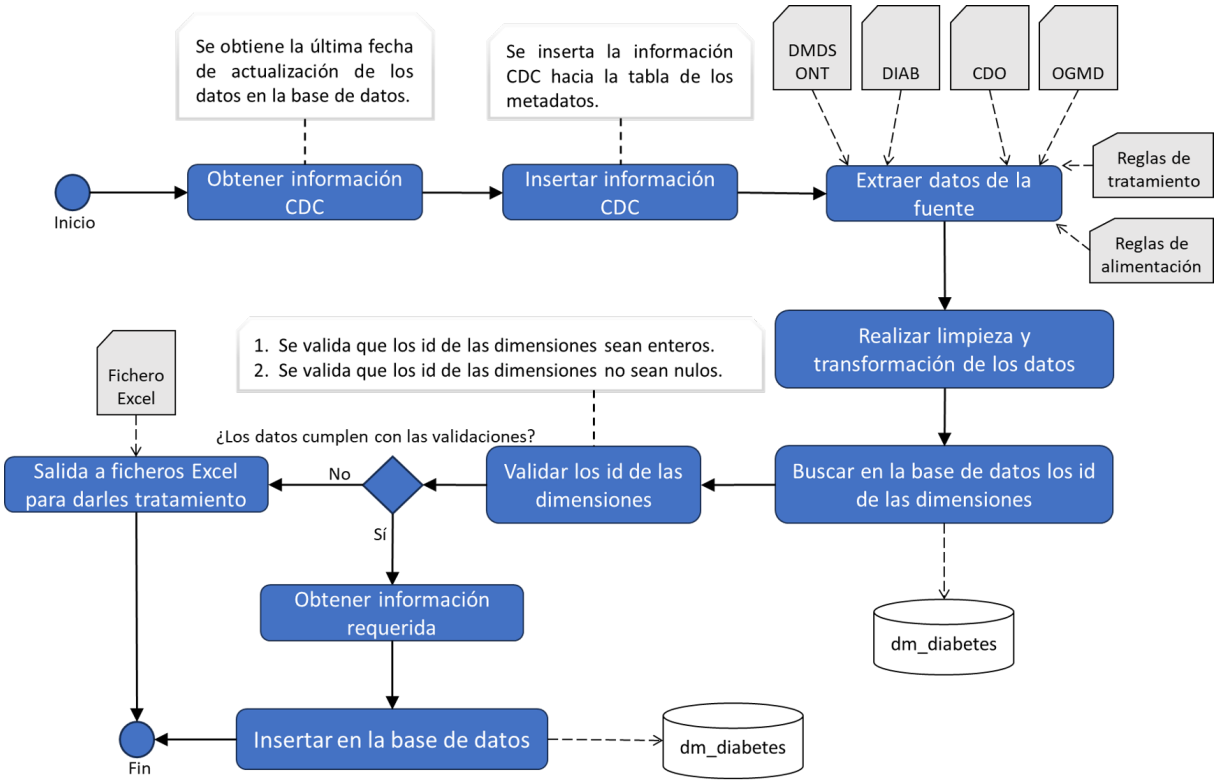


Figura 3.4: Diseño de las transformaciones para la carga de hechos.

3.1.2. Conversión de la base de conocimiento

Como se planteó a inicios del capítulo, una vez conformado el mercado de datos, es necesaria la conversión de este a una base de conocimiento, por lo que la definición de hechos y dimensiones fue un proceso crucial para el diseño de la base de conocimiento. Para modelar el conocimiento, se utiliza el concepto de ontología, el cual hace una representación formal de un conjunto de clases y las relaciones entre ellas. La ontología propuesta sigue la misma línea de los hechos y dimensiones definidos, los cuales fueron convertidos en clases.

Los componentes que conforman la ontología propuesta son:

- **Clases (conceptos)**: representan categorías de objetos o ideas en el dominio, e.g., `Paciente`, `Medicamentos`, `Tipo_diabetes`, etc.
- **Propiedades (atributos)**: son características que describen las clases, e.g., `nombre`, `edad`, `peso` y `altura` de la clase `Paciente`.
- **Instancias**: son ejemplos específicos de las clases, e.g., `Juan`, `30`, `62 kg`, `162 cm`.
- **Relaciones**: define las relaciones entre las clases, e.g., `recibe_Tratamiento` es la relación que asocia un `Paciente` con un `Tratamiento`.
- **Axiomas**: representan las reglas o restricciones en las que se definen cómo se utilizan las clases y sus propiedades, e.g., un axioma puede indicar que un paciente sólo puede tener un tratamiento a la vez, según el tipo de diabetes que presenta.

- **Jerarquías**: son las estructuras que organizan las clases por niveles de generalidad y especificidad, e.g., `Tratamiento` es la clase general y `Medicamentos` es una subclase.

La Tabla 3.3 muestra las clases, propiedades y jerarquías existentes en la ontología propuesta.

Tabla 3.3: Relación de clases, propiedades y jerarquías de la ontología

Componentes	Objetos	Descripción
Clases	`Alimentos`	Denota los alimentos que pueden ser consumidos por un paciente de diabetes.
	`Otras_enfermedades`	Describe otras enfermedades que puede tener un paciente de diabetes.
	`Paciente`	Representa al paciente en cuestión y describe propiedades como la edad, el peso, etc.
	`Sintomas_post`	Contiene los síntomas que puede tener un paciente que ya tiene diabetes.
	`Sintomas_prediabetes`	Define los síntomas que puede presentar una persona antes de ser diagnosticada con diabetes.
	`Test_diagnostico`	Representa las pruebas realizadas a un paciente antes y después de ser diagnosticado con diabetes.
	`Tipo_diabetes`	Define los diferentes tipos de diabetes existentes.
	`Tratamiento`	Se refiere a los tratamientos asociados a cada paciente.
	`Medicamentos`	Contiene los medicamentos asociados a la diabetes.
Propiedades	`formaParte`	Relaciona las clases "Medicamentos" y `Tratamientos`, e.g., Medicamento forma parte de un Tratamiento.
	`tiene_glucemia`	Asocia la clase `Paciente` y su propiedad de nivel de glucemia.
	`tiene_otras_enfermedades`	Relaciona las clases `Paciente` y `Otras_enfermedades`.
	`presenta_sintomas_post`	Relaciona las clases `Paciente` y `Sintomas_post`.
	`presenta_sintomas_pred`	Enlaza las clases `Paciente` y `Sintomas_prediabetes`.
	`tiene_testD`	Liga las clases `Paciente` y `Test_diagnostico`.
	`tiene_tipoD`	Asocia las clases `Paciente` y `Tipo_diabetes`.
	`tiene_genero`	Une la clase `Paciente` y su propiedad de género.
	`recibe_tratamiento`	Conecta las clases `Paciente` y `Tratamiento`.
	`tratam_depende_de`	Vincula las clases `Tratamiento`, `Medicamentos`, `Otras_enfermedades`, `Tipo_diabetes` y `Paciente`.
Jerarquías	`Tratamiento-Medicamentos`	La clase general es `Tratamiento` y una de sus subclases es `Medicamentos`.

3.2 Servicio de web semántica

El diseño del servicio de web semántica se centra en la integración de los datos relevantes, la creación de una estructura coherente y el desarrollo de mecanismos que permitan

la inferencia y recuperación eficiente de la información, proporcionando una plataforma que facilite el acceso a datos precisos y contextualizados.

3.2.1. Definición de requisitos

Requisitos funcionales (RF):

- RF-1 **Búsqueda en la base de conocimiento**: el sistema debe permitir la búsqueda de información sobre pacientes con diabetes dentro de una base de conocimiento interna.
- RF-2 **Búsqueda Web externa**: el sistema debe buscar en la Web si la información no está disponible internamente.
- RF-3 **Operaciones CRUD**: el sistema debe crear, leer, actualizar y eliminar datos en la base de conocimiento.

Requisitos no funcionales (RNF):

- RNF-1 **Disponibilidad y rendimiento**: el sistema de ser altamente disponible y responder de manera eficiente a las solicitudes.
- RNF-2 **Seguridad**: el sistema debe proteger los datos sensibles y cumplir con las normas de privacidad.
- RNF-3 **Interoperabilidad**: el sistema de ser capaz de trabajar con otros sistemas y plataformas.

3.2.2. Diseño de la API RESTful

En este punto se identifican los recursos que se usan en el desarrollo, i.e., datos que pueden ser identificados y gestionados dentro del sistema. Dichos datos se representan mediante URLs (*Uniform Resource Locators*) y se manipulan utilizando los métodos HTTP estándar GET, POST, PUT y DELETE. Para el servicio de web semántica orientado a la gestión de pacientes con diabetes, los recursos incluyen entidades como:

- Recursos:
  - /pacientes: recurso que gestiona la información del paciente al registrar sus datos.
  - /tratamientos: recurso que maneja los tratamientos asociados a un paciente.
  - /alimentación: recurso que se encarga de la alimentación que debe consumir un determinado paciente.
- Estructura de URLs y métodos HTTP:
  - GET/pacientes/{id}: obtiene información de un determinado paciente según su ID.
  - POST/pacientes: crea un nuevo paciente una vez que este introduce sus datos.

- PUT/pacientes/{id}: actualiza la información de un paciente según su ID.
- GET/tratamientos: obtiene los tratamientos asociados a un paciente.
- PUT/tratamientos{id}: actualiza los tratamientos de un paciente.
- GET/alimentos: obtiene los alimentos consumibles por un paciente.
- PUT/tratamientos{id}: actualiza los alimentos de un paciente.

3.2.3. Arquitectura del servicio de web semántica

Los componentes de la arquitectura son los siguientes:

- **Cliente**: para este caso, el cliente es el dispositivo del paciente de diabetes que consumirá el servicio de web semántica.
- **Servicio de web semántica con RESTful**: gestiona las solicitudes de los pacientes, interactúa con la base de conocimiento y, cuando es necesario, realiza las búsquedas en la Web para proporcionar información inexistente en la base de conocimiento.
- **Base de conocimiento**: almacena y gestiona el conocimiento relacionado con la diabetes de manera estructurada, utilizando ontologías para representar la información y permitir inferencias automáticas.
- **Web**: proporciona las respuestas a las solicitudes no encontradas en la base de conocimiento interna.

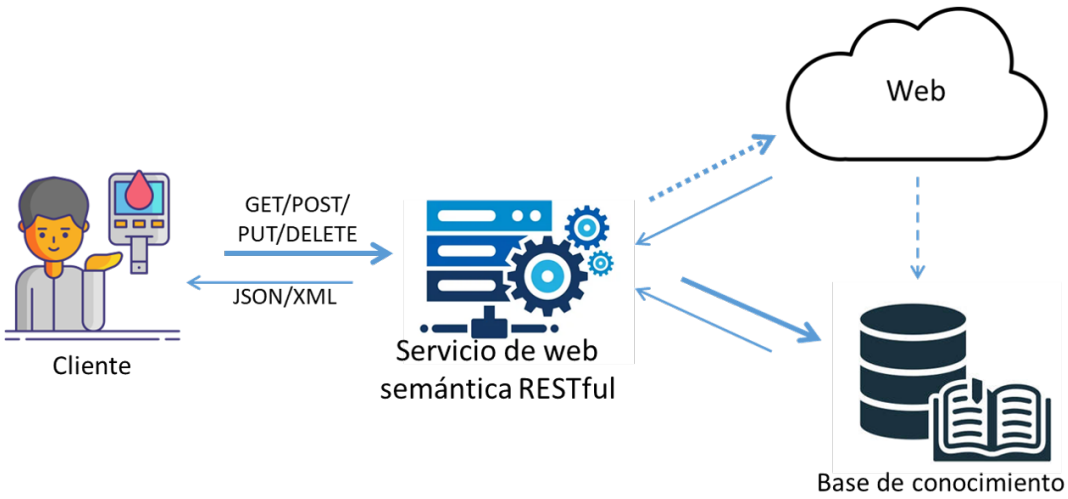


Figura 3.5: Arquitectura del servicio de web semántica.

Como se muestra en la Figura 3.5, la arquitectura del servicio de web semántica con la especificación de RESTful considera la integración con la base de conocimiento, ya que teniendo en cuenta una petición del paciente, dicho servicio realiza la búsqueda en la base de conocimiento y le regresa una repuesta con base en la solicitud realizada. Si la solicitud del paciente no tiene respuesta en la base de conocimiento, el servicio de web semántica realiza la búsqueda en la Web, procesa la información de manera que esta sea entendible por el paciente, devuelve la respuesta al paciente y nutre la base de conocimiento con la nueva información.

Capítulo 4

Implementación del sistema

En el presente capítulo se describe la implementación de los componentes de base de conocimiento y el servicio de web semántica. Primeramente, se explica la implementación del mercado de datos, haciendo énfasis en el proceso de almacenamiento e integración de los datos (cf. Sección 4.1). A continuación, se describe la construcción de la ontología mediante la creación de clases y subclases del dominio y la definición de sus relaciones e instancias; asimismo se detalla la traducción al formato OWL de la ontología y el desarrollo de consultas y reglas asociadas al dominio (cf. Sección 4.2). Por último, se explica la implementación del servicio de web semántica, ahondando en el procedimiento de búsqueda y en el procesamiento de la información (cf. Sección 4.3).

4.1 Implementación del mercado de datos de diabetes

La creación del mercado de datos de `Diabetes` es fundamental para la gestión eficiente de los datos recopilados. Este componente centraliza la información necesaria para el análisis y la toma de decisiones, permitiendo un acceso rápido y preciso a los datos relevantes. A continuación, se detallan los procesos que se llevan a cabo para la implementación del mercado de datos, destacando las estrategias utilizadas para garantizar su integridad y disponibilidad.

4.1.1. Proceso de almacenamiento

En la etapa de análisis y diseño del mercado de datos se definieron las tablas de hechos y dimensiones (cf. Sección 3.1.1). Estas están contenidas dentro de esquemas definidos para dicho mercado, los cuales permiten estructurar la información de manera eficiente.

Los esquemas definidos para el mercado de datos son los siguientes:

- `dm_diabetes`: contiene las tres tablas de hechos identificadas para el mercado de datos `Diabetes` (cf. Tabla 3.1 de la Sección 3.1.1).
- `dimension`: incluye las 10 dimensiones identificadas para el mercado de datos `Diabetes` (cf. Tabla 3.2 de la Sección 3.1.1).
- `dm_cdc`: contiene la tabla con información CDC (*Change Data Capture*).

Para lograr un mejor entendimiento entre las partes implicadas en el mercado de datos de `Diabetes`, se utilizan estándares de codificación, los cuales se muestran en la Tabla 4.1.

Tabla 4.1: Estándar de codificación del mercado de datos de `Diabetes`

Tipo de objeto	Función	Nomenclatura	Descripción
Esquemas	Dimensiones	`dimension`	Esquema donde se encuentran las tablas de dimensiones.
	Hechos	`dm_diabetes`	Esquema donde se encuentran las tablas de hechos.
	Metadatos	`dm_cdc`	Esquema donde se encuentra la tabla con información CDC.
Tablas	Dimensiones	`dim_[nombre]`	Tablas de dimensiones utilizadas como perspectivas de análisis.
	Hechos	`hech_[nombre]`	Tablas de hechos que definen las principales medidas o peticiones de un paciente.

Las herramientas utilizadas para el proceso de almacenamiento de los datos se describen a continuación:

- **Sistema Gestor de Bases de Datos (SGBD)**

  Un SGBD permite la administración de bases de datos, entre ellos se encuentran *Oracle Database, MySQL, PostgrSQL, Microsoft SQL Server*, entre otros.

  Para la administración de la base de datos propuesta se utiliza PostgreSQL, un sistema de bases de datos relacional de objetos. Además, es una herramienta de código abierto que posee características esenciales como la robustez, la facilidad de administración y la implementación de estándares. Utiliza multiprocesos en lugar de multihilos para garantizar la estabilidad del sistema, por lo que un fallo en alguno de sus procesos no afecta al resto y el sistema sigue en funcionamiento [45].

- **Administrador de Bases de Datos (ABD)**

  pgAdmin 4 v.8.3 es una herramienta de código abierto, rica en funciones para PostgreSQL, por lo cual garantiza una integración optimizada. Ofrece una interfaz intuitiva y amigable que permite la realización de tareas administrativas sin necesidad de conocer comandos SQL [43].

4.1.2. Proceso de integración de datos

Las herramientas para la extracción, la transformación y la carga (ETL) de datos facilitan los procesos de integración de datos desde diferentes fuentes de información y formatos. PDI (*Pentaho Data Integration*) o *Kittle* es una herramienta de la suite de Pentaho, de código abierto, aplicable a diferentes tipos de bases de datos como SQL Server, MySQL y PostgreSQL [33].

El proceso de ETL es la base fundamental de un almacén de datos, ya que mediante este, se extraen los datos de los sistemas fuentes, después se transforman para que cumplan con las restricciones de calidad de los mismos y finalmente se cargan en la base de datos para ser presentados a los usuarios finales.

Según Kimball y Caserta [31], el proceso de ETL consiste en:

1. **Extracción**: se refiere a la extracción de datos desde los sistemas de origen o fuentes, convirtiendo los datos a un formato preparado para el inicio del proceso de transformación.

2. **Transformación**: consiste en cualquier operación que se realice sobre los datos para que puedan ser cargados en el almacén de datos. En esta fase se aplica un conjunto de reglas o funciones sobre los datos que han sido extraídos para luego cargarlos en la base de datos destino.

3. **Carga**: se trata de almacenar los datos de la fase anterior (transformación) en la base de datos final.

Como se especificó en la Sección 3.1.1) del Capítulo 3, la fuente de datos utilizada se extrajo de ontologías relacionadas con diabetes, que están contenidas en el sitio BioPortal[4]. Estas ontologías se descargaron en formato *.json* y sirvieron de fuente de datos para las transformaciones de las tablas de hechos y dimensiones.

Las ontologías obtenidas del sitio BioPortal fueron las siguientes:

- Ontología de Diagnóstico y Soporte de Diabetes *Mellitus* (DMDSONT, en inglés *Diabetes Mellitus Diagnosis and Support Ontology*). DMDSONT cuenta con ocho clases y 71 instancias, en las que describe conceptos y relaciones asociadas a la diabetes *mellitus* [9].

- Ontología de la diabetes de BioMedBridges (DIAB, en inglés *BioMedBridges Diabetes Ontology*). DIAB cuenta con 375 conceptos divididos entre clases y subclases, donde describe la diabetes *mellitus* y los fenotipos asociados a ella. [7].

- Ontología china de la diabetes *mellitus* (CDO, en inglés *Chinese Diabetes Mellitus Ontology*). CDO está compuesta por 3752 clases que describen desde las manifestaciones clínicas de un paciente, hasta los medicamentos o productos biológicos, la estructura del cuerpo del paciente, entre otros términos [8].

- Ontología del trastorno del metabolismo de la glucosa (OGMD, en inglés *Ontology of Glucose Metabolism Disorder*). OGMD tiene un total de 134 clases que incluyen enfermedades, fenotipos, complicaciones de la diabetes, trastornos metabólicos, entre otros [10].

Como parte del proceso de ETL, se realizó la carga en la base de datos de las dimensiones y hechos. Como se ilustra en la Figura 4.1, el proceso de ETL de la dimensión `dim_test_diagnostico` comienza con la extracción de la fuente de datos (*.json* de las ontologías), una para cada ontología identificada. Luego se dividen las pruebas de diagnóstico y se colocan en una columna. A continuación, se agrega una columna con la descripción o traducción de las diferentes pruebas en términos entendibles por el paciente, se eliminan las columnas innecesarias y se asigna un ID secuencial a cada prueba. Por último, se insertan dichas pruebas en la base de datos destino (`dim_test_diagnostico`).

[4]Disponible en: *https://bioportal.bioontology.org*

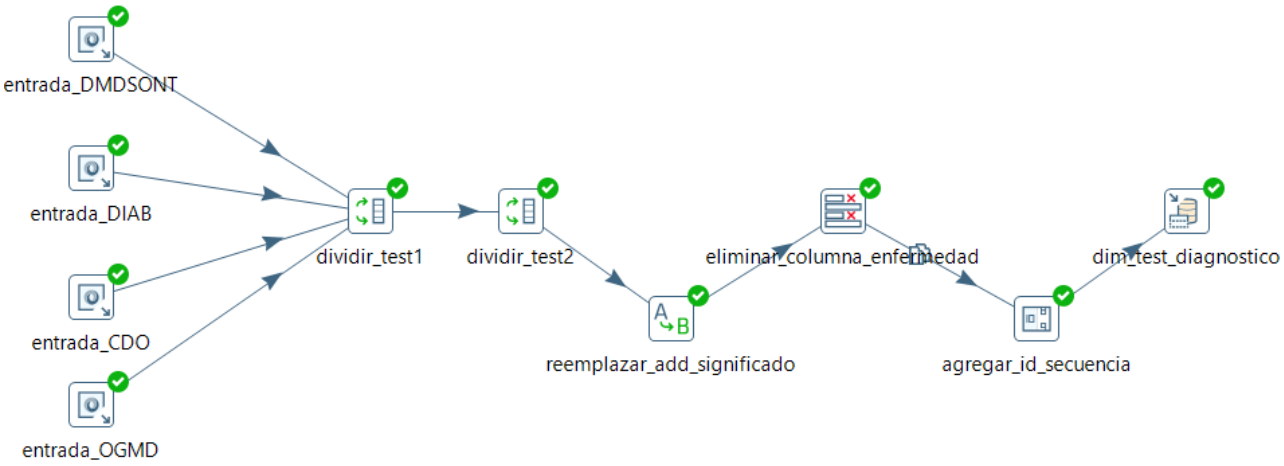


Figura 4.1: Proceso de integración de datos para la dimensión `dim_test_diagnostico`.

El hecho tratamiento (`hech_tratamiento`) almacena todas las posibles combinaciones de tratamientos definidos, teniendo en cuenta el paciente y sus características, como el peso, la altura, la edad, el género, el tipo de diabetes que presenta y otras enfermedades que pueda tener. La Figura 4.2 muestra la transformación para la carga de los datos del hecho `hech_tratamiento`. Antes de cargar los datos de la fuente, se obtiene la información CDC asociada a la fecha y el ID de la última actualización de la base de datos y se inserta dicha información en la tabla `cdc_hech_tratamiento`. Luego se comienza con la carga de la fuente de datos, se hacen las uniones de las tablas de los datos de la fuente a través del ID del paciente, se ordenan para unificarlas y se verifica que no hayan tuplas repetidas. A continuación, se seleccionan las columnas que se quieren mostrar y se realiza la búsqueda de los datos en la fuente de cada una de las dimensiones involucradas para establecer un tratamiento. Por último, se verifican los errores relativos a anomalías en los ID, como valores nulos o que no sean del tipo de datos enteros; de existir algún error se envía a un fichero Excel. Si los ID son correctos, se inserta la información en el hecho `hech_tratamiento`.

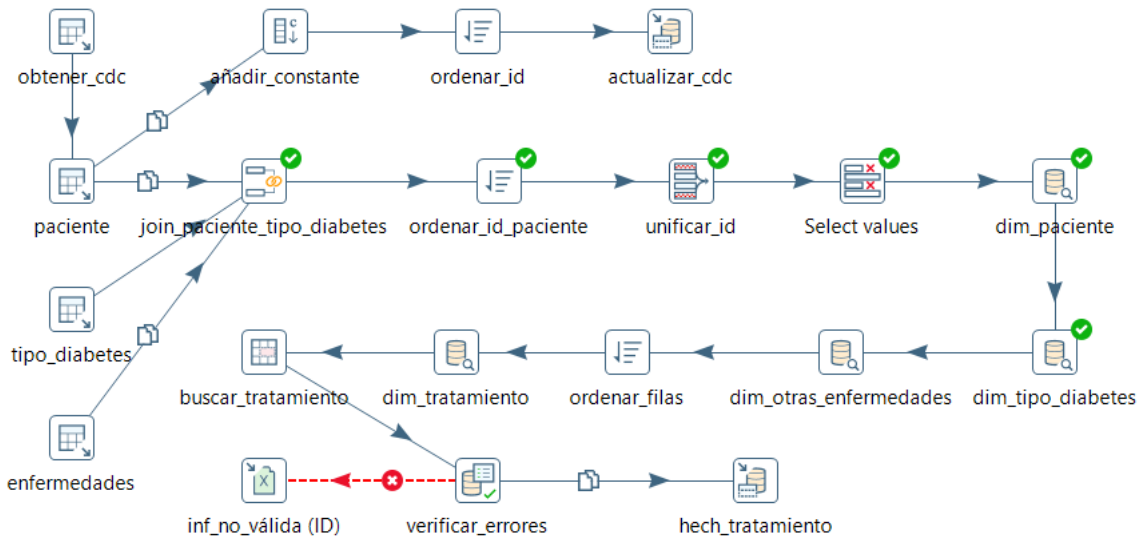


Figura 4.2: Proceso de integración de datos para el hecho `hech_tratamiento`.

4.2 Conversión del mercado de datos a la base de conocimiento

En la Sección 3.1.2 del Capítulo 3, se identificaron los conceptos asociados a la base de conocimiento, a través de las tablas de hechos y dimensiones definidos. Una vez pobladas las tablas de hechos, como se explicó en la Sección 4.1.2, estas sirven para definir las reglas que forman parte de la ontología que guiará la búsqueda en la base de conocimiento.

En el ámbito del modelado semántico y la gestión del conocimiento, existen diversas herramientas que facilitan la creación y la manipulación de ontologías. Una de estas herramientas es Protégé. la cual es de código abierto, utilizada para la creación, edición y visualización de ontologías. Presenta una interfaz gráfica intuitiva que facilita la creación de clases, propiedades, instancias y axiomas. Soporta múltiples formatos de ontologías como RDF/XML, OWL/XML, etc. Tiene la capacidad de inferencia, lo que permite razonar sobre la ontología para identificar inconsistencias, redundancias y relaciones implícitas [21].

4.2.1. Creación de clases y subclases

El primer paso para la concepción de la ontología consiste en la creación de las clases y subclases que representan los conceptos claves en el dominio de la diabetes. En este caso, se definen 26 conceptos divididos entre clases y subclases. La Tabla 4.2 muestra la relación entre las clases y subclases definidas.

Tabla 4.2: Clases y subclases de la ontología

Clases	Subclases
`Alimentos`	Frutas, Frutos secos, Granos integrales, Legumbres, Productos lácteos, Verduras
`Otras enfermedades`	Depresión y ansiedad, Enfermedad cardiovascular, Enfermedad periodontal, Enfermedad renal, Gastroparesia, Infección en la piel, Neuropatía diabética, Otra enfermedad, Presión arterial, Retinopatía diabética
`Paciente`	Género
`Síntomas post`	-
`Síntomas prediabetes`	-
`Test de diagnóstico`	-
`Tipo de diabetes`	-
`Tratamiento`	Medicamentos

4.2.2. Definición de propiedades y relaciones

Luego de ser creadas las clases y subclases, se definen las propiedades y relaciones entre ellas. Las propiedades de datos describen los atributos de las clases como se muestra en la Tabla 4.3.

Tabla 4.3: Propiedades de las clases

Clases	Propiedades
`Alimentos`	nombre, calorías, carbohidratos, proteínas, grasas y vitaminas
`Otras enfermedades`	nombre, categoría
`Paciente`	nombre, edad, peso, género y altura
`Síntomas post`	nombre, descripción y frecuencia
`Síntomas prediabetes`	nombre, descripción y frecuencia
`Test de diagnóstico`	nombre, tipo, descripción y rango normal
`Tipo de diabetes`	nombre, descripción, edad de aparición, factor de riesgo y tratamientos comunes
`Tratamiento`	tipo de paciente, otras enfermedades, medicamento, modalidad, frecuencia, dosis y descripción
`Medicamentos`	nombre, dosis, forma farmacéutica, vía de administración, fecha de aprobación, fecha de caducidad y fabricante

Las propiedades de objetos establecen relaciones entre las clases. La Tabla 4.4 muestra varios ejemplos de estas relaciones.

Tabla 4.4: Relaciones entre clases

Relaciones	Descripción
`recibe_tratamiento`	El paciente recibe tratamiento.
`presenta_sintomas_pred`	El paciente presenta síntomas de prediabetes.
`tratam_depende_de`	Un tratamiento depende de los medicamentos para controlar la diabetes, el paciente, el tipo de diabetes diagnosticada y otras enfermedades que presente.

La representación de las relaciones entre las clases y las propiedades, de manera gráfica, se muestra en la Figura 4.3

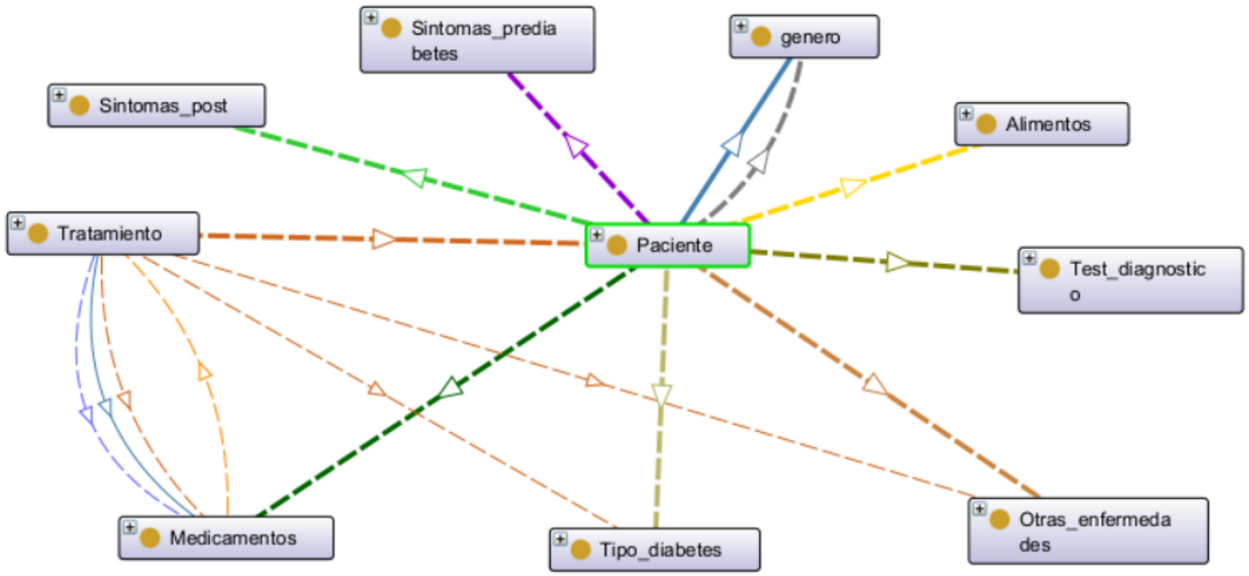


Figura 4.3: Grafo reducido de la ontología propuesta.

4.2.3. Traducción de la ontología al formato OWL

La ontología desarrollada se tradujo a OWL, permitiendo su representación formal y estandarizada. OWL facilita la interoperabilidad y el uso de la ontología en diversas aplicaciones.

Primeramente, se configura el entorno para definir la ontología, estableciendo el contexto y especificando los prefijos necesarios para que el documento OWL sea válido y comprensible. A continuación, se definen las bases sobre las que se construye la ontología, permitiendo el uso de vocabularios estándar como RDF, RDFS, OWL y SWRL (Código 4.1).

```
1   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
3     xmlns:owl="http://www.w3.org/2002/07/owl#"
4     xmlns:swrl="http://www.w3.org/2003/11/swrl#"
5     xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
6     xmlns:ex="http://example.org/ontology#">
7
```

```
8     <owl:Ontology rdf:about="http://example.org/ontology"/>
```

Código 4.1: Definición de prefijos y vocabularios estándares de la ontología.

Posteriormente, se declaran las clases y subclases con el elemento *Class*. El Código 4.2 muestra un ejemplo de declaración de las clases `Paciente`, `Tratamiento`, `Tipo_diabetes`, `Otras_enfermedades` y la subclase `Medicamentos`.

```
1     <owl:Class rdf:about="&ex;Paciente"/>
2     <owl:Class rdf:about="&ex;Tratamiento"/>
3     <owl:Class rdf:about="&ex;Medicamentos">
4         <rdfs:subClassOf rdf:resource="&ex;Tratamiento"/>
5     </owl:Class>
6     <owl:Class rdf:about="&ex;Tipo_diabetes"/>
7     <owl:Class rdf:about="&ex;Otras_enfermedades"/>
```

Código 4.2: Declaración de clases y subclases de la ontología.

Las propiedades de estas clases se declararon a través de *DatatypeProperty*. Los Códigos 4.3 y 4.4, muestran las declaraciones de las propiedades de las clases `Paciente` y `Tratamiento`, respectivamente. Asimismo, las relaciones entre clases se declaran con la palabra clave *ObjectProperty* como se muestra en el Código 4.5.

```
1     <owl:DatatypeProperty rdf:about="&ex;nombre">
2         <rdfs:domain rdf:resource="&ex;Paciente"/>
3         <rdfs:range rdf:resource="&swrlb;string"/>
4     </owl:DatatypeProperty>
5
6     <owl:DatatypeProperty rdf:about="&ex;edad">
7         <rdfs:domain rdf:resource="&ex;Paciente"/>
8         <rdfs:range rdf:resource="&swrlb;integer"/>
9     </owl:DatatypeProperty>
10
11    <owl:DatatypeProperty rdf:about="&ex;peso">
12        <rdfs:domain rdf:resource="&ex;Paciente"/>
13        <rdfs:range rdf:resource="&swrlb;float"/>
14    </owl:DatatypeProperty>
15
16    <owl:DatatypeProperty rdf:about="&ex;genero">
17        <rdfs:domain rdf:resource="&ex;Paciente"/>
18        <rdfs:range rdf:resource="&swrlb;string"/>
19    </owl:DatatypeProperty>
20
21    <owl:DatatypeProperty rdf:about="&ex;altura">
22        <rdfs:domain rdf:resource="&ex;Paciente"/>
23        <rdfs:range rdf:resource="&swrlb;float"/>
24    </owl:DatatypeProperty>
```

Código 4.3: Declaración de las propiedades de la clase `Paciente`.

```
1     <owl:DatatypeProperty rdf:about="&ex;medicamento">
2         <rdfs:domain rdf:resource="&ex;Tratamiento"/>
3         <rdfs:range rdf:resource="&xsd;string"/>
4     </owl:DatatypeProperty>
5
```

```
6     <owl:DatatypeProperty rdf:about="&ex:modalidad">
7         <rdfs:domain rdf:resource="&ex;Tratamiento"/>
8         <rdfs:range rdf:resource="&xsd;string"/>
9     </owl:DatatypeProperty>
10
11    <owl:DatatypeProperty rdf:about="&ex:dosis">
12        <rdfs:domain rdf:resource="&ex;Tratamiento"/>
13        <rdfs:range rdf:resource="&xsd;string"/>
14    </owl:DatatypeProperty>
15
16    <owl:DatatypeProperty rdf:about="&ex:frecuencia">
17        <rdfs:domain rdf:resource="&ex;Tratamiento"/>
18        <rdfs:range rdf:resource="&xsd;string"/>
19    </owl:DatatypeProperty>
20
21    <owl:DatatypeProperty rdf:about="&ex:descripcion">
22        <rdfs:domain rdf:resource="&ex;Tratamiento"/>
23        <rdfs:range rdf:resource="&xsd;string"/>
24    </owl:DatatypeProperty>
```

Código 4.4: Declaración de las propiedades de la clase `Tratamiento`.

```
1     <owl:ObjectProperty rdf:about="&ex;tiene_tipoD">
2         <rdfs:domain rdf:resource="&ex;Paciente"/>
3         <rdfs:range rdf:resource="&ex;Tipo_diabetes"/>
4     </owl:ObjectProperty>
5
6     <owl:ObjectProperty rdf:about="&ex:recibe_tratamiento">
7         <rdfs:domain rdf:resource="&ex;Paciente"/>
8         <rdfs:range rdf:resource="&ex;Tratamiento"/>
9     </owl:ObjectProperty>
10
11    <owl:ObjectProperty rdf:about="&ex:tratam_depende_de">
12        <rdfs:domain rdf:resource="&ex;Tratamiento"/>
13        <rdfs:range>
14            <owl:Class>
15                <owl:unionOf rdf:parseType="Collection">
16                    <rdf:Description rdf:about="&ex;Paciente"/>
17                    <rdf:Description rdf:about="&ex;Tipo_diabetes"/>
18                    <rdf:Description rdf:about="&ex;Medicamentos"/>
19                    <rdf:Description rdf:about="&ex;Otras_enfermedades"/>
20                </owl:unionOf>
21            </owl:Class>
22        </rdfs:range>
23    </owl:ObjectProperty>
```

Código 4.5: Declaración de las relaciones entre las clases.

4.2.4. Creación de instancias

La creación de instancias implica agregar datos específicos para representar casos concretos dentro del dominio de la diabetes. Para ello, se crearon instancias de pacientes con sus respectivos diagnósticos y características, así como del resto de las clases definidas y las relaciones entre ellas. Los Códigos 4.6 y 4.7, muestran un ejemplo de la creación de las instancias para las clases `Paciente` y `Tratamiento`, respectivamente.

```
<Paciente rdf:ID="Paciente_1">
    <tiene_edad rdf:datatype="&xsd;integer">45</tiene_edad>
    <tiene_genero rdf:datatype="&xsd;string">Masculino</tiene_genero>
    <tiene_peso rdf:datatype="&xsd;float">70.5</tiene_peso>
    <tiene_altura rdf:datatype="&xsd;float">1.65</tiene_altura>
    <tiene_TipoD rdf:resource="#Diabetes_Tipo_1"/>
    <presenta_otras_enfer rdf:resource="#Insuficiencia_renal"/>
    <!-- <recibe_tratamiento rdf:resource="#Tratamiento_1"/> -->
</Paciente>

<Paciente rdf:ID="Paciente_2">
    <tiene_edad rdf:datatype="&xsd;integer">62</tiene_edad>
    <tiene_genero rdf:datatype="&xsd;string">Femenino</tiene_genero>
    <tiene_peso rdf:datatype="&xsd;float">87.0</tiene_peso>
    <tiene_altura rdf:datatype="&xsd;float">1.63</tiene_altura>
    <tiene_TipoD rdf:resource="#Diabetes_Tipo_2"/>
    <presenta_otras_enfer rdf:resource="#Hipertension"/>
    <!-- <recibe_tratamiento rdf:resource="#Tratamiento_2"/> -->
</Paciente>

<Paciente rdf:ID="Paciente_3">
    <tiene_edad rdf:datatype="&xsd;integer">29</tiene_edad>
    <tiene_genero rdf:datatype="&xsd;string">Femenino</tiene_genero>
    <tiene_peso rdf:datatype="&xsd;float">55.4</tiene_peso>
    <tiene_altura rdf:datatype="&xsd;float">1.59</tiene_altura>
    <tiene_TipoD rdf:resource="#Diabetes_Gestacional"/>
    <!-- <presenta_otras_enfer rdf:resource=""/> -->
    <!-- <recibe_tratamiento rdf:resource="#Tratamiento_3"/> -->
</Paciente>

<Paciente rdf:ID="Paciente_4">
    <tiene_edad rdf:datatype="&xsd;integer">55</tiene_edad>
    <tiene_genero rdf:datatype="&xsd;string">Masculino</tiene_genero>
    <tiene_peso rdf:datatype="&xsd;float">65.0</tiene_peso>
    <tiene_altura rdf:datatype="&xsd;float">1.68</tiene_altura>
    <tiene_TipoD rdf:resource="#Diabetes_Tipo_Mody"/>
    <presenta_otras_enfer rdf:resource="#Infarto_miocardio"/>
    <!-- <recibe_tratamiento rdf:resource="#Tratamiento_4"/> -->
</Paciente>
```

Código 4.6: Ejemplos de la creación de instancias para la clase `Paciente`.

```
<Tratamiento rdf:ID="Tratamiento_1">
    <tratam_depende_de rdf:resource="#Diabetes_Tipo_2"/>
    <tratam_depende_de rdf:resource="#Enfermedad_Renal"/>
    <tratam_depende_de rdf:resource="#Presion_Arterial"/>
    <tratam_depende_de rdf:resource="#Insulina"/>
    <tiene_modalidad rdf:datatype="&xsd;string">Inyectable</tiene_modalidad
        >
    <tiene_frecuencia rdf:datatype="&xsd;string">1 vez al día</
        tiene_frecuencia>
    <tiene_dosis rdf:datatype="&xsd;string">10 ml</tiene_dosis>
    <tiene_descripción rdf:datatype="&xsd;string">Insulina, inyectable, 1
        vez al día, 10 ml</tiene_descripción>
</Tratamiento>

<Tratamiento rdf:ID="Tratamiento_2">
```

```
13      <tratam_depende_de rdf:resource="#Diabetes_Tipo_1"/>
14      <tratam_depende_de rdf:resource="#Arritmias"/>
15      <tratam_depende_de rdf:resource="#Metformina"/>
16      <tiene_modalidad rdf:datatype="&xsd;string">Tabletas</tiene_modalidad>
17      <tiene_frecuencia rdf:datatype="&xsd;string">1 después de cada comida</
            tiene_frecuencia>
18      <tiene_dosis rdf:datatype="&xsd;string">1 tableta</tiene_dosis>
19      <tiene_descripción rdf:datatype="&xsd;string">Metformina, tabletas, 1
            después de cada comida, 1 tableta</tiene_descripción>
20  </Tratamiento>
21
22  <Tratamiento rdf:ID="Tratamiento_3">
23      <tratam_depende_de rdf:resource="#Diabetes_Gestacional"/>
24      <tratam_depende_de rdf:resource="#Dieta_Saludable"/>
25      <tiene_descripción rdf:datatype="&xsd;string">Dieta saludable</
            tiene_descripción>
26  </Tratamiento>
```

Código 4.7: Ejemplos de la creación de instancias para la clase `Tratamiento`.

4.2.5. Desarrollo de consultas y reglas de inferencia

Se desarrollaron consultas SPARQL y reglas de inferencia en OWL para extraer conocimiento de la ontología y probar su correcta implementación. Las consultas SPARQL permitieron recuperar información específica y las reglas de inferencia permitieron derivar nuevo conocimiento. El Código 4.8, se presenta una consulta SPARQL que determina el tratamiento asociado a un paciente con base en el tipo de diabetes, otras enfermedades que presenta y la edad del paciente.

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX owl: <http://www.w3.org/2002/07/owl#>
4  PREFIX ex: <http://example.org/ontology#>
5
6  SELECT ?paciente ?tratamiento ?medicamento ?modalidad ?dosis ?frecuencia
7  WHERE {
8      ?paciente rdf:type ex:Paciente .
9      ?paciente ex:tiene_tipoD ?Tipo_diabetes .
10     ?paciente ex:presenta_otras_enfer ?Otras_enfermedades .
11     ?paciente ex:tiene_edad ?edad .
12
13     ?tratamiento ex:tratam_depende_de ?Tipo_diabetes .
14     ?tratamiento ex:tratam_depende_de ?Otras_enfermedades .
15
16     ?tratamiento ex:medicamento ?medicamento .
17     ?tratamiento ex:modalidad ?modalidad .
18     ?tratamiento ex:dosis ?dosis .
19     ?tratamiento ex:frecuencia ?frecuencia .
20 }
```

Código 4.8: Consulta SPARQL para determinar el tratamiento de un paciente.

Las reglas de inferencia en OWL se escriben utilizando SWRL y permiten deducir automáticamente relaciones adicionales entre las clases de la ontología. Por ejemplo, con base en la información sobre el tipo de diabetes, la edad, el peso y otras características

del paciente, como otras enfermedades que presente, se puede inferir qué tratamiento es el más adecuado para él.

El Código 4.9, muestra un ejemplo de regla de inferencia implementada, la cual determina el tratamiento adecuado para un paciente, considerando el tipo de diabetes y su rango de edad con respecto al conjunto de tratamientos. La regla # 1 se aplica a un `Paciente` que tiene diabetes tipo 1 y cuya edad está dentro de un rango específico y considera un `Tratamiento` que es adecuado para dicho tipo de diabetes y rango de edad. Si el paciente cumple con estas condiciones, entonces se infiere que el `Paciente` debe recibir el `Tratamiento` correspondiente.

```
<swrl:Imp rdf:about="#Regla1">
  <swrl:body rdf:parseType="Collection">
    <!-- Paciente con diabetes tipo 1 y edad ?edad1 -->
    <swrl:IndividualPropertyAtom>
      <swrl:propertyPredicate rdf:resource="&ex;tiene_tipoD"/>
      <swrl:argument1 rdf:resource="?paciente"/>
      <swrl:argument2 rdf:resource="&ex;diabetes_tipo1"/>
    </swrl:IndividualPropertyAtom>
    <swrl:DataPropertyAtom>
      <swrl:propertyPredicate rdf:resource="&ex:tiene_edad"/>
      <swrl:argument1 rdf:resource="?paciente"/>
      <swrl:argument2 rdf:resource="?edad1"/>
    </swrl:DataPropertyAtom>
    <!-- Tratamiento para diabetes tipo 1, con rangos de edad ?edadMin1 y ?
        edadMax1 -->
    <swrl:IndividualPropertyAtom>
      <swrl:propertyPredicate rdf:resource="&ex:tratam_depende_de"/>
      <swrl:argument1 rdf:resource="?tratamiento"/>
      <swrl:argument2 rdf:resource="&ex;diabetes_tipo1"/>
    </swrl:IndividualPropertyAtom>
    <swrl:DataPropertyAtom>
      <swrl:propertyPredicate rdf:resource="&ex:edadMin"/>
      <swrl:argument1 rdf:resource="?tratamiento"/>
      <swrl:argument2 rdf:resource="?edadMin1"/>
    </swrl:DataPropertyAtom>
    <swrl:DataPropertyAtom>
      <swrl:propertyPredicate rdf:resource="&ex:edadMax"/>
      <swrl:argument1 rdf:resource="?tratamiento"/>
      <swrl:argument2 rdf:resource="?edadMax1"/>
    </swrl:DataPropertyAtom>
    <!-- Condiciones de comparación de edad -->
    <swrl:BuiltinAtom>
      <swrl:builtin rdf:resource="&swrlb;greaterThanOrEqual"/>
      <swrl:arguments rdf:parseType="Collection">
        <rdf:Description rdf:about="?edad1"/>
        <rdf:Description rdf:about="?edadMin1"/>
      </swrl:arguments>
    </swrl:BuiltinAtom>
    <swrl:BuiltinAtom>
      <swrl:builtin rdf:resource="&swrlb;lessThanOrEqual"/>
      <swrl:arguments rdf:parseType="Collection">
        <rdf:Description rdf:about="?edad1"/>
        <rdf:Description rdf:about="?edadMax1"/>
      </swrl:arguments>
    </swrl:BuiltinAtom>
  </swrl:body>
```

```
46    <swrl:head rdf:parseType="Collection">
47      <!-- Asignación del tratamiento al paciente -->
48      <swrl:IndividualPropertyAtom>
49        <swrl:propertyPredicate rdf:resource="&ex:recibe_tratamiento"/>
50        <swrl:argument1 rdf:resource="?paciente"/>
51        <swrl:argument2 rdf:resource="?tratamiento"/>
52      </swrl:IndividualPropertyAtom>
53    </swrl:head>
54  </swrl:Imp>
```

Código 4.9: Regla # 1 de `Tratamiento` definida con base en el tipo de diabetes y la edad del paciente.

La regla # 2 que se muestra en el Código 4.10, determina el tratamiento adecuado para un paciente que presente una enfermedad específica. La regla se aplica a un `Paciente` que presenta enfermedad renal y considera un `Tratamiento` que es conveniente para dicha enfermedad. Si el `Paciente` cumple con esta condición, entonces se infiere que el `Paciente` debe recibir el `Tratamiento` correspondiente.

```
1  <swrl:Imp rdf:about="#Regla2">
2    <swrl:body rdf:parseType="Collection">
3      <rdf:Description rdf:about="#Paciente">
4        <rdf:type rdf:resource="&ex;Paciente"/>
5        <ex:presenta_otras_enfer rdf:resource="&ex;enfermedad_renal"/>
6      </rdf:Description>
7      <rdf:Description rdf:about="#Tratamiento">
8        <rdf:type rdf:resource="&ex;Tratamiento"/>
9        <ex:tratam_depende_de rdf:resource="&ex;enfermedad_renal"/>
10     </rdf:Description>
11   </swrl:body>
12   <swrl:head rdf:parseType="Collection">
13     <rdf:Description rdf:about="#Paciente">
14       <ex:recibe_tratamiento rdf:resource="#Tratamiento"/>
15     </rdf:Description>
16   </swrl:head>
17 </swrl:Imp>
```

Código 4.10: Regla # 2 de `Tratamiento` con base en la presencia de una enfermedad.

4.3 Implementación del servicio de web semántica

Como se mencionó en la Sección 2.1 del Capítulo 2, un servicio de web semántica es una aplicación diseñada para establecer comunicación con otro programa, en este caso, entre el paciente de diabetes y la base de conocimiento o la Web.

Para la implementación del servicio de web semántica, especialmente en el contexto de desarrollo de ontologías y servicios semánticos, existen varias herramientas y *frameworks* disponibles que facilitan el proceso:

- Python Flask es un *frameworks* web ligero y flexible para Python. Es adecuado para el desarrollo de servicios web RESTful mediante rutas y métodos HTTP y puede integrarse fácilmente con bibliotecas de manejo de datos y ontologías [18].
- Apache Jena es un marco de web semántica de Java, de código abierto, que proporciona una API para la extracción de datos y escritura en grafos RDF. Para este

caso, teniendo en cuenta que el servicio de web semántica está implementado con el lenguaje Python, se utiliza Apache Jena Fuseki, que proporciona una interfaz RESTful para consultas SPARQL y actualizaciones, lo que permite interactuar con la base de datos RDF a través de solicitudes HTTP desde la aplicación Flask [19].

4.3.1. Búsqueda y procesamiento de la información

El servicio de web semántica, bajo la especificación de RESTful, emplea un procedimiento estructurado para manejar solicitudes de datos. Este procedimiento garantiza que se proporcione la información más actualizada y relevante, ya sea desde la base de conocimiento interna, implementada en la Sección 4.2, o mediante la búsqueda en la Web externa. Los pasos del proceso de búsqueda de información en el servicio de web semántica se describen a continuación (cf. Figura 4.4):

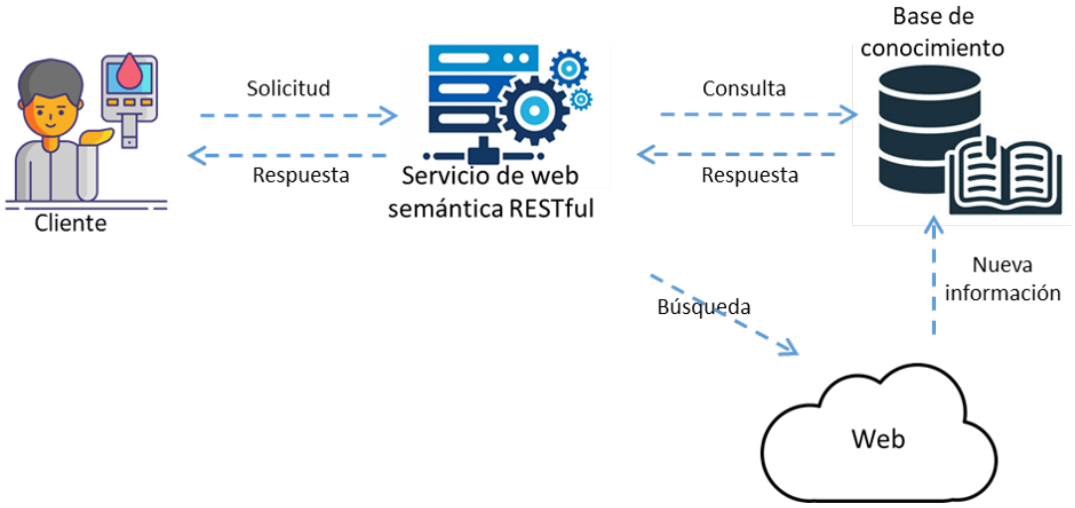


Figura 4.4: Proceso de búsqueda en el servicio de web semántica.

1. **Recepción de la solicitud del paciente**: el paciente envía una solicitud al servicio de web semántica a través de una interfaz de usuario. La solicitud puede incluir preguntas sobre el manejo de la diabetes, síntomas específicos, información sobre su medicación, alimentación, etc.

2. **Consulta a la base de conocimiento**: el servicio de web semántica recibe la solicitud y primero consulta la base de conocimiento para buscar la respuesta adecuada. La base de conocimiento está estructurada utilizando tecnologías semánticas como OWL y RDF. La consulta se realiza utilizando SPARQL para buscar la información en la ontología. Un ejemplo de consulta SPARQL se muestra en el Código 4.8.

3. **Procesamiento de la respuesta**: si la base de conocimiento contiene la respuesta a la solicitud del paciente, esta se procesa para que sea comprensible y útil para el paciente. La respuesta se estructura y se envía al paciente a través del servicio de web semántica.

4. **Búsqueda en la Web**: si la base de conocimiento no contiene la respuesta a la solicitud del paciente, el servicio de web semántica realiza la búsqueda en la Web. La información obtenida de la Web se procesa para generar una respuesta entendible por el paciente.

5. **Nutrición de la base de conocimiento**: la nueva información obtenida de la Web se valida y se estructura para integrarse en la base de conocimiento. Esta integración se realiza mediante la actualización de la ontología y el almacenamiento de la nueva información en formato RDF.

Estructura del proyecto

Para tener un mejor control de los archivos de implementación y mantener el proyecto organizado y modular, se estructuró como sigue (cf. Figura 4.5):

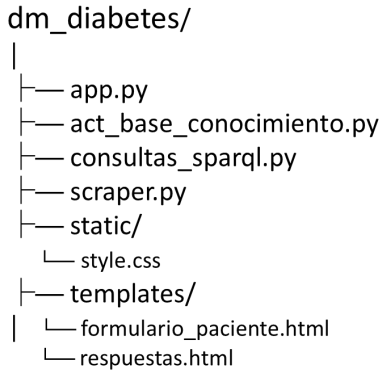


Figura 4.5: Estructura del proyecto

- `app.py`: contiene la configuración del servidor Flask y la definición de los *endpoints* del servicio de web semántica.
- `act_base_conocimiento`: maneja las actualizaciones del grafo de conocimiento con la nueva información obtenida.
- `consultas_sparql.py`: implementa las funciones necesarias para realizar consultas SPARQL a la base de conocimiento.
- `scraper.py`: define las funciones para realizar web *scraping* y obtener información adicional de la Web.
- `static/style.css`: contiene los estilos CSS para cada interfaz de usuario que visualiza el paciente.
- `templates/formulario_paciente.html`: proporciona una interfaz de usuario donde el paciente introduce sus datos y selecciona el tipo de búsqueda.
- `templates/respuestas.html`: proporciona una interfaz de usuario al paciente para visualizar los resultados a sus solicitudes.

Configuración del servidor Flask

El servidor Flask maneja las solicitudes de los pacientes y coordina las consultas a la base de conocimiento y el web *scraping*. Para ello se definen los *endpoints* o rutas específicas a las que los pacientes pueden acceder. En este contexto, los recursos o servicios que se ponen a disposición del paciente son aquellos que permiten acceder a información relevante, como recomendaciones de tratamientos o datos obtenidos a través de web *scraping*, dependiendo de la petición realizada por el paciente.

El *endpoint* `/query` permite a los pacientes realizar consultas 'GET' para obtener información sobre su tratamiento, alimentación o diagnóstico con base en los parámetros introducidos por ellos. La solicitud incluye dichos parámetros y el ID del paciente como argumentos de consulta. El servidor Flask primero intenta recuperar la información de la base de conocimiento; si no la encuentra, realiza una búsqueda en la Web y actualiza la base de conocimiento con la nueva información obtenida. Un ejemplo de solicitud sería `GET /query?patient_id=123`, cuya respuesta se muestra en el Código 4.11

```
{
        "tratamiento_1":
        [
            "Insulina",
            "Inyectable",
            "1 vez al día",
            "10 ml",
            "Insulina, inyectable, 1 vez al día, 10 ml"
        ]
    }
```

Código 4.11: Respuesta a una solicitud en formato *.json*.

La implementación de la función `/query` se muestra en el Código 4.12. Dicha implementación consiste en la recepción de la solicitud del paciente, donde el servidor recibe la solicitud 'GET' con el ID del paciente. Luego se ejecuta la consulta en la base de conocimiento en busca de la información. Si la petición del paciente se encuentra en la base de conocimiento, se le envía la respuesta en formato *.json*. Si dicha petición no se encuentra disponible en la base de conocimiento, se realiza la búsqueda en la Web a través de web *scraping*. La nueva información se integra en la base de conocimiento y, de igual manera, se envía la respuesta al paciente.

```
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/query', methods=['GET'])
def query():
    # Recepción de la solicitud
    patient_id = request.args.get('patient_id')

    # Consulta a la base de conocimiento
    result = consulta_sparql(patient_id)

    # Búsqueda en la Web (scraping)
    if not result:
        result = scrape_web()

        # Actualización de la base de conocimiento
        actualizar_base(patient_id, result)

    return jsonify(result)  # Respuesta en formato .json
```

Código 4.12: Implementación de la función `/query()`.

Consultas SPARQL

El servicio de web semántica recibe la solicitud y primero consulta en la base de conocimiento para encontrar la respuesta adecuada, la cual está estructurada utilizando tecnologías semánticas como OWL y RDF (cf. Sección 4.2). La consulta se realiza mediante SPARQL para buscar información en la ontología.

Un ejemplo de la implementación del módulo `consultas_sparql.py` se muestra en el Código 4.13, donde se carga la base de conocimiento en formato RDF. Luego se construye la consulta SPARQL para obtener la información solicitada por el paciente. Para ello, se tiene en cuenta los parámetros relacionados con las características del paciente, como el tipo de diabetes y las enfermedades que presenta. Posteriormente, se ejecuta la consulta y se recopilan los resultados obtenidos para mostrarlos al paciente.

```
1  from rdflib import Graph, URIRef
2
3  def consulta_sparql(patient_id):
4      g = Graph()
5      g.parse('dm_diabetes.rdf')
6
7      query = f"""
8      PREFIX ex: <http://example.org/ontology#>
9      SELECT ?paciente ?tratamiento ?medicamento ?modalidad ?dosis ?
           frecuencia
10     WHERE {{
11         ?paciente ex:patient_id "{patient_id}" .
12         ?paciente rdf:type ex:Paciente .
13         ?paciente ex:tiene_tipoD ?Tipo_diabetes .
14         ?paciente ex:presenta_otras_enfer ?Otras_enfermedades .
15         ?paciente ex:tiene_edad ?edad .
16
17         ?tratamiento ex:tratam_depende_de ?Tipo_diabetes .
18         ?tratamiento ex:tratam_depende_de ?Otras_enfermedades .
19
20         ?tratamiento ex:medicamento ?medicamento .
21         ?tratamiento ex:modalidad ?modalidad .
22         ?tratamiento ex:dosis ?dosis .
23         ?tratamiento ex:frecuencia ?frecuencia .
24     }}
25     """
26
27     results = g.query(query)
28
29     # Procesar los resultados en un formato más fácil de usar
30     treatments = []
31     for row in results:
32         treatment_info = {
33             "paciente": str(row.paciente),
34             "tratamiento": str(row.tratamiento),
35             "medicamento": str(row.medicamento),
36             "modalidad": str(row.modalidad),
37             "dosis": str(row.dosis),
38             "frecuencia": str(row.frecuencia)
39         }
40         treatments.append(treatment_info)
41
```

```
42     return treatments if treatments else None
```

Código 4.13: Implementación de la función `consulta_sparql()`.

Web *Scraping*

El proceso de web *scraping* se utiliza para la búsqueda de la información que no se encuentra en la base de conocimiento. Para la implementación del módulo de web *scraping* del servicio de web semántica se utiliza el «PROYECTO de Norma Oficial Mexicana PROY-NOM-015-SSA2-2018, para la prevención, detección, diagnóstico, tratamiento y control de la Diabetes *Mellitus*» [46], la cual establece los procedimientos y medidas necesarias para la prevención, la identificación, el diagnóstico y control de la prediabetes y diabetes *mellitus*.

El Código 4.14 muestra la función, de manera general, de la web *scraping*. Primeramente, se solicita y recibe el contenido HTML de la página especificada. Luego, dicho contenido se analiza usando la biblioteca *BeautifulSoup*. Seguidamente, se busca y extrae texto de todos los elementos de la página que contienen la clase o parámetro de búsqueda y se guarda en una lista.

```
1  import requests
2  from bs4 import BeautifulSoup
3
4  def scrape_web():
5      treatments = []
6
7      # Scraping de la página DOF
8      url = 'https://www.dof.gob.mx/nota_detalle.php?codigo=5521405&fecha
           =03/05/2018#gsc.tab=0'
9      response = requests.get(url)
10
11     # Verificar si la solicitud fue exitosa
12     if response.status_code == 200:
13         soup = BeautifulSoup(response.content, 'html.parser')
14
15         # Buscar y añadir la información de tratamientos
16         for treatment in soup.find_all(class_='treatment'):
17             treatments.append(treatment.get_text(strip=True))
18
19     else:
20         print(f"Error al acceder a la página: {response.status_code}")
21
22     return treatments
```

Código 4.14: Implementación de la función `scrape_web()`.

Actualización de la base de conocimiento

La nueva información obtenida de la Web es estructurada para su integración en la base de conocimiento. Esta integración se realiza mediante la actualización de la ontología y el almacenamiento de la nueva información en formato RDF. Como muestra el Código 4.15, primeramente se carga el archivo RDF que contiene la base de conocimiento. Luego se añaden las nuevas *triples* al grafo RDF existente, que consiste en tripletas de datos compuestas por un sujeto, un predicado y un objeto, y se describen los hechos o relaciones entre los recursos. Por último, se guarda el archivo RDF actualizado. Este paso asegura

que la nueva información obtenida, a través de web *scraping*, se integre en la base de conocimiento.

```
from rdflib import Graph, URIRef, Literal, Namespace

def actualizar_base(patient_id, treatments):
    g = Graph()
    g.parse('dm_diabetes.rdf')

    EX = Namespace('http://example.org/')
    patient = URIRef(EX[patient_id])

    # Verificar si el paciente existe
    if (patient, None, None) in g:
        for treatment in treatments:
            g.add((patient, EX.hasTreatment, Literal(treatment)))
    else:
        print(f"El paciente con ID '{patient_id}' no se encontró en la base
            de conocimiento.")

    # Serializar y guardar los cambios en el archivo RDF
    g.serialize('dm_diabetes.rdf', format='turtle')
```

Código 4.15: Implementación de la función `actualizar_base()`.

Visualización de resultados

Para tener un resultado exacto en tiempo real y, al mismo tiempo, que el paciente introduzca sus datos, se implementó una interfaz de usuario a modo de formulario en la que el paciente introduce sus datos como el nombre, la edad, el peso, la altura, el nivel de glucemia actual o el promedio que suele tener, el género, el tipo de diabetes y otras enfermedades que padece, como se muestra en la Figura 4.6. Esta información es almacenada en la base de datos con un ID de paciente, mismo ID que se utiliza luego para mostrar, según los parámetros registrados del paciente en cuestión, el tratamiento, la alimentación o el diagnóstico asociados a este paciente.

Figura 4.6: Interfaz de usuario mostrando el formulario del paciente.

Por otra parte, para visualizar las respuestas a las solicitudes del paciente, se implementó una interfaz de usuario en la que se devuelve un mensaje flotante con la respuesta asociada a la solicitud del paciente, como se muestra en la Figura 4.7.

Figura 4.7: Ejemplos de respuestas sobre tratamientos para pacientes.

Capítulo 5

Pruebas

En este capítulo se explicará el proceso de pruebas de integración de los componentes desarrollados para detectar problemas y corregirlos, comenzando por las pruebas al mercado de datos `Diabetes` (cf. Sección 5.1), así como las pruebas realizadas a la ontología (cf. Sección 5.2) y pruebas de prototipo con usuarios finales (cf. Sección 5.3) para medir la carga de trabajo y de percepción y usabilidad estética de la aplicación.

5.1 Pruebas al mercado de datos `Diabetes`

Durante el desarrollo de un producto de *software*, es importante la realización de pruebas al mismo en aras de determinar su calidad. Estas pruebas se realizan desde el inicio del *software* hasta que llegue a manos del cliente, esperando que cumpla con sus expectativas. Para llevar a cabo las pruebas del mercado de datos de `Diabetes` fue utilizado el modelo V, como se muestra en la Figura 5.1. Este modelo facilita la realización de pruebas a un determinado sistema, mediante la ejecución de procesos que describen los pasos durante el ciclo de vida de un proyecto. La letra "V" significa verificación y validación [51]. Además, este modelo es una representación de dos cascadas relacionadas con un vértice común en la codificación, donde la cascada izquierda muestra las actividades de análisis y diseño del mercado de datos de `Diabetes` y la cascada derecha representa las actividades relacionadas con el aseguramiento de la calidad mediante los tipos de pruebas aplicadas.

Las pruebas a realizar plantean [51]:

- **Pruebas unitarias**: constituyen la primera fase de las pruebas dinámicas y se realizan a cada componente o módulo del *software* de manera individual.
- **Pruebas de integración**: consisten en comprobar el sistema como un todo para determinar la integración de los componentes del mismo, así como evaluar su funcionalidad y desempeño.
- **Pruebas de sistema**: se llevan a cabo para comprobar el funcionamiento del sistema y validar el cumplimiento de los objetivos planteados.
- **Pruebas de aceptación**: se realizan directamente por los clientes para validar los requerimientos definidos.

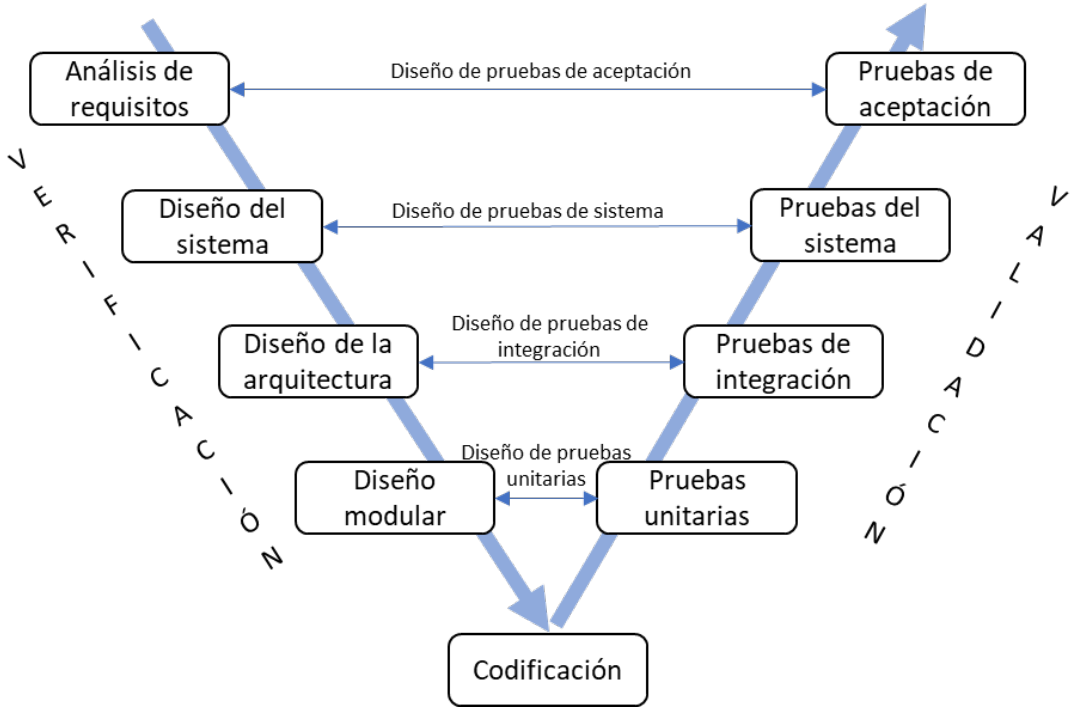


Figura 5.1: Modelo V.

5.1.1. Herramienta para la aplicación de las pruebas

Los casos de prueba son un conjunto de guías que incluyen pasos y resultados esperados durante la ejecución de una prueba funcional del mercado de datos. Describen los pasos detallados que serán seguidos para verificar el funcionamiento del sistema [24]. Para el mercado de datos `Diabetes` fueron diseñados tres Casos de Prueba (CP), que corresponden a tres de las áreas de análisis definidas. La Tabla 5.1, muestra el diseño propuesto.

Tabla 5.1: Diseño de los Casos de prueba.

Escenario	Descripción	Variable de entrada	Variable de salida	Respuesta del sistema
CP-1: Muestra los tratamientos.	Muestra los tratamientos asociados al paciente.	▪ Paciente (edad, peso, altura) ▪ Medicamentos ▪ Otras enfermedades ▪ Tipo de diabetes ▪ Nivel de glucemia	Listado de tratamientos	El sistema muestra los tratamientos que cumplan o estén en el rango de los datos del paciente.
CP-2: Muestra los alimentos.	Muestra los alimentos que puede consumir el paciente según el nivel de glucemia actual	▪ Paciente (edad, peso, altura) ▪ Nivel de glucemia	Listado de alimentos	El sistema muestra los alimentos que el paciente puede consumir.
CP-3: Muestra el diagnóstico.	Muestra el tipo de diagnóstico que debe realizarse a un paciente.	▪ Paciente (edad, peso, altura) ▪ Tipo de diabetes	Listado de diagnósticos	El sistema muestra el tipo de diagnóstico que debe realizarse a un paciente, según sus características y tipo de diabetes.

5.1.2. Resultados de las pruebas

El mercado de datos `Diabetes` se implementó hasta la fase de carga de los datos en la base de datos, sin el proceso de visualización, ya que los datos se visualizan a través del servicio de web semántica, con las consultas SPARQL realizadas sobre la ontología. Las pruebas realizadas arrojaron los siguientes resultados:

Pruebas unitarias e integración

Con este tipo de pruebas se verificó que cada componente individual del mercado de datos `Diabetes`, como las funciones de transformación de datos, las operaciones de inserción y actualización en la base de datos y las funciones de consultas, funcionen correctamente de manera aislada, así mismo cuando se integren entre sí para detectar y corregir las No Conformidades (NC).

A continuación, se listan las No Conformidades detectadas, las cuales fueron resueltas:

1. Redefinir las tablas de hechos de manera que respondan a las necesidades de los pacientes.

- **Estado inicial**: las tablas de hechos originales no estaban completamente alineadas con los requerimientos específicos de los pacientes, lo que limitaba la capacidad de generar los resultados adecuados.
- **Acción realizada**: se llevó a cabo la redefinición de las tablas de hechos para incluir métricas más relevantes, como el seguimiento de niveles de glucosa, tratamientos recibidos y pruebas de diagnóstico. También se agregaron dimensiones adicionales para proporcionar un contexto más enriquecido, como `dim_tipo_diabetes` y `dim_medicamentos`.
- **Estado final**: las nuevas tablas definidas permiten una mayor flexibilidad en los análisis de los datos. Además, están optimizadas para responder a las consultas específicas de los usuarios.

2. Optimizar la utilización de componentes en la implementación de las transformaciones.
   - **Estado inicial**: las transformaciones de datos no estaban aprovechando al máximo los componentes disponibles, lo que generaba ineficiencias en el proceso de ETL.
   - **Acción realizada**: se revisaron los flujos de transformación, utilizando funciones más eficientes. Además, se mejoraron los *scripts* de transformación para reducir el tiempo de ejecución.
   - **Estado final**: la utilización de componentes más eficientes han reducido los tiempos de procesamiento de datos, permitiendo actualizaciones más frecuentes.
3. Comprobar que los datos cargados coincidan con los de la fuente.
   - **Estado inicial**: no había un proceso estructurado para verificar que los datos cargados en el mercado de datos coincidieran exactamente con los datos de la fuente, lo que introducía el riesgo de inconsistencias.
   - **Acción realizada**: se implementaron Casos de Prueba para verificar la integridad y exactitud de los datos cargados. Estos Casos de Prueba consisten en realizar comparaciones automatizadas entre los datos en la fuente y los datos cargados en el mercado de datos.
   - **Estado final**: los Casos de Prueba permitieron detectar y corregir errores en la carga de datos, asegurando que los datos almacenados sean consistentes y fiables.

Pruebas del sistema

Las pruebas de sistema realizadas consistieron en verificar tanto el proceso de ETL, como el sistema en general con los datos cargados. Se realizaron dos iteraciones en las cuales fueron detectadas un total de cuatro No Conformidades en la primera y dos No Conformidades en la segunda. La Tabla 5.2, muestra el desglose de las No Conformidades detectadas según su complejidad.

Tabla 5.2: Resultados de las pruebas de sistema.

Iteraciones	Cantidad de NC	Complejidad			NC resueltas
		Alta	Media	Baja	
Iteración 1	4	2	1	1	4
Iteración 2	2	1	1	0	2

Primera iteración

1. NC-1: Errores en la carga de datos (complejidad alta).
   - **Descripción**: durante el proceso de carga de datos en el mercado de datos, se detectaron errores en el mapeo de los campos. Algunos datos no se insertaban correctamente en las tablas correspondientes, lo que generaba inconsistencias.
   - **Resolución**: se revisaron las transformaciones aplicadas en la etapa de transformación del ETL, ajustando el mapeo de campos y agregando validaciones para detectar errores antes de la carga.
2. NC-2: Inconsistencias en las transformaciones de datos (complejidad alta).
   - **Descripción**: se encontraron problemas en la transformación de algunos valores, como la conversión de unidades o el manejo de datos nulos, lo que afectaba la precisión en la base de datos.
   - **Resolución**: se implementaron reglas de transformación adicionales para manejar casos especiales, como valores faltantes.
3. NC-3: Problemas de rendimiento en el procesamiento de grandes volúmenes de datos (complejidad media).
   - **Descripción**: al incrementar el volumen de datos, el tiempo de ejecución del proceso de ETL se incrementaba significativamente, afectando el rendimiento del mercado de datos.
   - **Resolución**: se optimizó el uso de componentes de las transformaciones y se realizaron ajustes en las consultas SQL para manejar la eficiencia. Además, se implementó la carga incremental de datos en lugar de una carga completa en cada iteración.
4. NC-4: Errores en la definición de las tablas de hechos (complejidad baja).
   - **Descripción**: se detectaron errores en la estructura de las tablas de hechos, lo que dificultaba el almacenamiento adecuado de los datos agregados. Algunos campos no estaban bien definidos y faltaban índices necesarios.
   - **Resolución**: se revisó el modelado de las tablas de hechos, corrigiendo la definición de los campos y agregando índices para manejar la velocidad de consultas.

Segunda iteración

1. NC-5: Problemas con la integración de los datos de múltiples fuentes (complejidad alta).

- **Descripción**: se detectaron errores al combinar datos provenientes de diferentes fuentes en la etapa de integración, lo que generaba duplicados y registros inconsistentes.
- **Resolución**: se implementaron reglas para eliminar duplicados, con el fin de fusionar correctamente los registros provenientes de diversas fuentes.

2. NC-6: Inconsistencias en la carga de datos históricos (complejidad media).

   - **Descripción**: al procesar datos históricos, se encontraron problemas al cargar información que no cumplía con las estructuras actuales de las tablas, lo que causaba errores de inserción.
   - **Resolución**: se realizaron ajustes en el proceso de ETL para adaptar la estructura de los datos históricos antes de la carga, asegurando su compatibilidad con las tablas actuales.

Pruebas de aceptación

Para este caso no se realizó pruebas de aceptación, ya que el mercado de datos `Diabetes` solo llega hasta el proceso de carga de los datos y conformación de la base de conocimiento. Este tipo de pruebas se realiza en las Secciones 5.3.1 y 5.3.2, donde se explican las pruebas de carga de trabajo y pruebas de percepción y usabilidad estética, respectivamente, que se aplican a un conjunto de usuarios.

5.2 Validación y pruebas a la ontología propuesta

El proceso de validación y pruebas de una ontología es crucial para garantizar su funcionalidad y precisión. Incluye un conjunto de pasos que se realizan utilizando herramientas específicas y técnicas manuales para asegurar que la ontología cumpla con sus objetivos y proporcione resultados fiables y precisos. Estas pruebas incluyen la verificación de la coherencia, la validación de los datos, así como las pruebas de consulta y las de inferencia, como se describe a continuación.

Verificación de coherencia

En este punto se asegura que la ontología no contenga contradicciones lógicas. Para este caso, se utiliza la herramienta HermiT, un razonador de ontologías que determina la consistencia de estas. Es de código abierto y está publicado bajo la licencia GPL (*GNU Lesser General Public License*) [12].

Utilizando HermiT dentro del entorno Protégé, se realizaron varias iteraciones de razonamiento. En cada iteración se evaluó la jerarquía de clases, las relaciones entre propiedades de objeto y datos, así como las afirmaciones de individuos. Con estas iteraciones se determinó que la ontología propuesta no presenta problemas de coherencia en cuanto a la definición de las clases ni subclases, ni en la lógica de sus relaciones. En la primera iteración se detectaron errores de inconsistencia en la jerarquía de las clases y cardinalidad de propiedades entre clases, como se muestra en la Figura 5.2. En la segunda iteración, el razonador detectó errores de individuos equivalentes e inconsistencias en las restricciones de datos. La Figura 5.3 muestra el resultado de esta iteración.

Primera iteración

- **Inconsistencia en la jerarquía de clases**: se detectó que la clase `Medicamentos` tiene restricciones contradictorias. Por ejemplo, se define como subclase de `Tratamiento` y al mismo tiempo como una clase independiente de `Tratamiento`, lo que genera conflicto lógico.
- **Cardinalidad de la propiedad**: se especificó en la ontología que un `Paciente` debe tener exactamente un `Tratamiento`, pero algunos individuos tienen múltiples instancias asociadas a la propiedad `recibe_tratamiento`, violando la restricción de cardinalidad.
- **Resolución de errores**: para el error de jerarquía de clases, se revisó la estructura de la jerarquía para que la clase `Tratamiento` no esté en conflicto con la clase `Medicamento`. Además, se ajustaron las restricciones de la propiedad `recibe_tratamiento` y se corrigieron los datos de los individuos para que cumplan con las restricciones establecidas.

```
INFO  11:32:32  -------------------- Running Reasoner --------------------
INFO  11:32:32  Pre-computing inferences:
INFO  11:32:32      - class hierarchy
INFO  11:32:32      - object property hierarchy
INFO  11:32:32      - data property hierarchy
INFO  11:34:10      - class assertions
INFO  11:34:45      - object property assertions
INFO  11:35:30      - same individuals
INFO  11:36:25      - verifying consistency of all classes and properties
WARNING  11:36:30  Inconsistency detected in class hierarchy:
                  - Conflict between classes `Tratamiento` and `Medicamento`
WARNING  11:36:31  Cardinality constraint violation:
                  - `Paciente` has multiple `recibe_tratamiento` instances
INFO  11:36:32  Ontologies processed in 3 minutes and 50 seconds by HermiT
```

Figura 5.2: Proceso de razonamiento utilizando HermiT en Protégé (primera iteración).

Segunda iteración

- **Individuos considerados como equivalentes**: se identificaron dos individuos (`PacienteA` y `PacienteB`) marcados como equivalentes, debido a axiomas añadidos que indican que ambos tienen las mismas propiedades y valores de datos. Este resultado no era el esperado, ya que se suponía que representaban entidades diferentes.
- **Inconsistencia en las restricciones de datos**: la propiedad `tiene_glucemia` tiene una restricción de rangos de valores positivos, sin embargo, se detectaron individuos con valores negativos en dicha propiedad, lo cual viola las restricciones de datos.
- **Resolución de errores**: se ajustaron los axiomas que causaron la equivalencia no deseada, añadiendo restricciones para diferenciar mejor a los individuos `PacienteA` y `PacienteB`. Por otra parte, se corrigieron los datos de los individuos que tenían valores de `tiene_glucemia` negativos y se añadieron validaciones en la ontología para evitar que se ingresen valores fuera del rango permitido en futuras actualizaciones.

```
INFO  16:00:10  --------------------- Running Reasoner ---------------------
INFO  16:00:10  Pre-computing inferences:
INFO  16:00:10      - class hierarchy
INFO  16:01:30      - object property hierarchy
INFO  16:02:50      - data property hierarchy
INFO  16:04:10      - class assertions
INFO  16:05:30      - object property assertions
INFO  16:07:00      - same individuals
INFO  16:08:20      - verifying consistency of all classes and properties
WARNING  16:08:25  Equivalence issue detected between individuals:
                  - `PacienteA` and `PacienteB` marked as equivalent
WARNING  16:08:26  Inconsistency detected in data property:
                  - Negative values found for property `dosis`
INFO  16:09:37  Ontologies processed in 5 minutes and 27 seconds by HermiT
```

Figura 5.3: Proceso de razonamiento utilizando HermiT en Protégé (segunda iteración).

Validación de datos

La validación de los datos confirma que las instancias y las relaciones definidas en la ontología sean correctas y completas. Para ello, una vez creadas dichas instancias y relaciones, se verifica de manera manual que cada instancia tenga las propiedades y valores adecuados, según las especificaciones de la ontología. Así mismo, se asegura que los datos cumplan con las restricciones y axiomas definidos.

1. Verificación de propiedades y valores de las instancias:
    - Para cada instancia creada, se revisaron de forma manual las propiedades asociadas, confirmando que los valores asignados cumplieran con el tipo y rango esperado.
    - Por ejemplo, si una instancia de `Paciente` tiene la propiedad `tiene_edad`, se validó que el valor de esta propiedad fuera un número entero positivo dentro de un rango adecuado (entre 0 y 120).
    - También se comprobó que las propiedades obligatorias, definidas como requeridas en la ontología, estuvieran presentes en todas las instancias correspondientes.
2. Confirmación de la coherencia de las relaciones:
    - Las relaciones entre instancias, como las propiedades de objetos, se verificaron para asegurar que reflejen las conexiones establecidas en el diseño conceptual de la ontología.
    - Por ejemplo, para la relación `recibe_tratamiento` entre un `Paciente` y un `Tratamiento`, se confirmó que sólo instancias válidas de las clases `Paciente` y `Tratamiento` participaran en esta relación.
    - Esto incluyó revisar que no existieran relaciones contradictorias, como que un `Paciente` tuviera un `Tratamiento` que, según las restricciones, no debería aplicar.

3. Validación de restricciones y axiomas:

   - Se realizaron verificaciones para garantizar que los datos cumplieran con axiomas y restricciones definidos en la ontología. Esto incluyó las siguientes validaciones:
     - **Cardinalidad**: se validó que las propiedades que requerían un número específico de valores (un `Paciente` sólo puede tener un ID único) cumplían con estas restricciones.
     - **Disyunción**: se revisaron las clases disjuntas para evitar que una instancia se asignara incorrectamente a dos clases excluyentes.
     - **Tipo de datos y propiedades**: se comprobó que las propiedades de datos, como `tiene_altura`, siguieran un formato adecuado y que los tipos de datos correspondieran con los valores asignados.

4. Revisión de consistencia lógica:

   - Utilizando el razonador HermiT, se corrieron inferencias automáticas para identificar posibles inconsistencias lógicas. Esto permitió detectar errores de definición o de instancia que no siempre son evidentes en la validación manual.

Pruebas de consultas

Estas pruebas sirven para evaluar la precisión y eficiencia de las consultas SPARQL desarrolladas. Este punto consiste en escribir y ejecutar consultas SPARQL sobre la ontología para obtener los datos deseados. Se comparan los datos obtenidos con los resultados esperados para asegurarse de que las consultas devuelvan la información correcta y así medir la exactitud de las consultas. Para comparar los datos obtenidos, también se utilizó el modelo de consultas SQL, para verificar que a través de ambas consultas se obtuvieran los mismos datos.

Los Códigos 5.1 y 5.2, muestran un ejemplo de consultas SPARQL y SQL, respectivamente, donde se extrae información de tratamientos en función de los parámetros de un paciente. Las Figuras 5.4 y 5.5, muestran los resultados obtenidos de dichas consultas.

```
PREFIX : <http://www.example.org/ontology#>

  SELECT ?paciente ?edad ?peso ?Tipo_diabetes ?Otras_enfermedades
       ?Medicamento ?dosis ?modalidad ?frecuencia ?descripcion
  WHERE {
    ?paciente a :Paciente ;
         :edad ?edad ;
         :peso ?peso ;
         :tiene_tipoD ?Tipo_diabetes ;
         :presenta_otras_enfermed ?Otras_enfermedades ;
         :recibe_tratamiento ?Tratamiento .

    ?Tratamiento a :Tratamiento ;
          :tiene_dosis ?dosis ;
          :tiene_modalidad ?modalidad ;
          :tiene_frecuencia ?frecuencia ;
          :tiene_descripción ?descripcion ;
          :tiene_medicamento ?Medicamento .
```

```
19
20      # Filtrar los valores deseados para ?edad y ?peso
21      FILTER (?edad >= 31 && ?edad <= 42)
22      FILTER (?peso >= 63 && ?peso <= 71)
23      FILTER (?Tipo_diabetes = "Diabetes Tipo 2")
24      FILTER (?Otras_enfermedades = "Enfermedad renal")
25    }
```

Código 5.1: Consulta SPARQL ejecutada para extraer información de tratamientos en función de los parámetros del paciente.

Paciente	Edad	Peso	Tipo de Diabetes	Otras Enfermedades	Medicamento	Dosis	Modalidad	Frecuencia	Descripción
Paciente 1	31-42	63-71	Diabetes Tipo 2	Enfermedad renal	Insulina rapida	9ml	Inyectable	1 vez al día	Insulina rapida, Inyectable, 1 vez al día, 9ml

Figura 5.4: Resultados de la consulta SPARQL para determinar el tratamiento de un paciente.

```
1   SELECT
2     tratamiento->'tratamiento'->>'dosis' AS dosis,
3     tratamiento->'tratamiento'->>'modalidad' AS modalidad,
4     tratamiento->'tratamiento'->>'frecuencia' AS frecuencia,
5     tratamiento->'tratamiento'->>'descripcion' AS descripcion,
6     tratamiento->'tratamiento'->>'medicamento' AS medicamento
7   FROM
8     hech_tratamiento
9   WHERE
10    (tratamiento->'parametros'->>'tipo_diabetes') = 'Diabetes Tipo 2'
11    AND (tratamiento->'parametros'->>'edad') = '31-42'
12    AND (tratamiento->'parametros'->>'peso') = '63-71'
13    AND (tratamiento->'parametros'->'enfermedades') @> '["Enfermedad renal
        "]'::jsonb;
```

Código 5.2: Consulta SQL ejecutada para extraer información de tratamientos en función de los parámetros del paciente.

	medicamento text	modalidad text	dosis text	frecuencia text	descripcion text
1	Insulina rapida	Inyectable	9ml	1 vez al día	Insulina rapida, Inyectable, 1 vez al día, 9ml

Figura 5.5: Resultados de la consulta SQL para determinar el tratamiento de un paciente.

Asimismo, los Códigos 5.3 y 5.4, muestran un ejemplo de consultas SPARQL y SQL, respectivamente, donde se extrae información de los alimentos que debe consumir un paciente en función de sus valores de glucemia. Las Figuras 5.6 y 5.7, muestran los resultados obtenidos de dichas consultas.

```
1 PREFIX : <http://www.example.org/ontology#>
2
3 SELECT ?paciente ?nombre_alimento ?tipo_diabetes ?nivel_glucemia ?
      descripcion_alimento
4 WHERE {
5   ?paciente a :Paciente ;
```

```
6               :tiene_tipoD ?tipo_diabetes ;
7               :nivel_glucemia ?nivel_glucemia ;
8               :debe_consumir ?Alimentos .
9
10  ?Alimentos a :Alimentos ;
11              :tiene_nombre ?nombre_alimento ;
12              :tiene_descripcion ?descripcion_alimento .
13
14  # Filtrar por el nivel de glucemia deseado y tipo de diabetes del
        paciente
15  FILTER (?nivel_glucemia <= "130"^^xsd:integer)
16  FILTER (?tipo_diabetes = "Diabetes Tipo 2")
17 }
```

Código 5.3: Consulta SPARQL ejecutada para extraer información de los alimentos en función de los parámetros del paciente.

Paciente	Tipo de Diabetes	Nivel de Glucemia	Alimentos	Descripción
Paciente 1	Diabetes Tipo 2	130	Frutos secos, verdudas	Frutos secos ricos en grasas saludables. Verdura con bajo índice glucémico

Figura 5.6: Resultados de la consulta SPARQL para determinar los alimentos que debe consumir un paciente.

```
1  SELECT
2      alimento->'alimento'->>'nombre' AS nombre,
3      tipo_alimento,
4      alimento->'alimento'->>'descripcion' AS descripcion,
5      alimento->'parametros'->>'tipo_diabetes' AS tipo_diabetes,
6      alimento->'parametros'->>'nivel_glucemia_min' AS nivel_glucemia_min,
7      alimento->'parametros'->>'nivel_glucemia_max' AS nivel_glucemia_max
8  FROM
9      hech_alimentacion
10 WHERE
11     (alimento->'parametros'->>'tipo_diabetes') = 'Diabetes Tipo 2'
12     AND (CAST(alimento->'parametros'->>'nivel_glucemia_min' AS INTEGER) <=
           130)
13     AND (CAST(alimento->'parametros'->>'nivel_glucemia_max' AS INTEGER) >=
           130);
```

Código 5.4: Consulta SQL ejecutada para extraer información de los alimentos en función de los parámetros del paciente.

	nombre text	tipo_alimento character varying	descripcion text	tipo_diabetes text	nivel_glucemia_min text	nivel_glucemia_max text
1	Manzana	Fruta	Fruta rica en fibra y baja en azúcar	Diabetes Tipo 2	100	150
2	Avena	Cereal	Cereal integral que ayuda a controlar la glucosa	Diabetes Tipo 2	90	140

Figura 5.7: Resultados de la consulta SQL para determinar los alimentos que debe consumir un paciente.

Pruebas de inferencia

Las pruebas de inferencia comprueban la exactitud de las reglas de inferencia y los resultados derivados de ellas. Para ello, se verifica la correctitud de las inferencias generadas por las reglas establecidas, de manera que cumplan y coincidan con las expectativas.

Para validar este tipo de prueba se verificó que las reglas estuvieran bien definidas. Ejemplo de ello, con base al ejemplo anterior (ver Figura 5.1), si se tiene una regla que infiere un tratamiento para pacientes con "Diabetes Tipo 2", que también tengan "Enfermedad renal", se debe mostrar únicamente el tratamiento que cumpla con las condiciones antes mencionadas.

5.3 Pruebas a la aplicación

En esta sección, se presentan los resultados de las pruebas realizadas en la aplicación. Estas pruebas incluyeron pruebas de carga de trabajo (ver Sección 5.3.1) y pruebas de percepción y usabilidad estética (ver Sección 5.3.2). Para llevar a cabo estas pruebas, se convocaron a nueve usuarios, entre ellos, un Médico General Integral (MGI), que para esta ocasión fungió como paciente, pero también evaluó que las reglas establecidas para el diagnóstico, el tratamiento y la alimentación del paciente, estén correctamente definidas y ocho usuarios con diabetes, seleccionados según el tipo de diabetes, edad, género y otras enfermedades que padecen.

5.3.1. Pruebas de carga de trabajo

Las pruebas para evaluar la carga de trabajo se realizaron utilizando el cuestionario NASA-TLX (*NASA Task Load Index*) (ver Apéndice A), el cual evalúa de manera subjetiva la carga de trabajo que experimentan los usuarios durante su interacción con sistemas interactivos [1]. Esta herramienta es utilizada para medir la carga de trabajo percibida por los pacientes al interactuar con el servicio de web semántica. Para ello, se evalúan seis dimensiones definidas por preguntas [28]:

- **Demanda mental**: ¿Cuánta actividad mental y perceptiva requirió usted para comprender el funcionamiento de la aplicación?
- **Demanda física**: ¿Cuánta actividad física requirió usted para interactuar con la aplicación?
- **Demanda temporal**: ¿Cuánta presión de tiempo sintió usted debido al ritmo al que se procesa una petición en la aplicación?
- **Desempeño general**: ¿Qué tan exitoso fue usted al realizar la tarea? ¿Qué tan satisfecho estuvo con su desempeño?
- **Nivel de frustración**: ¿Qué tan irritado, estresado y molesto versus contento, relajado y complaciente se sintió usted durante la tarea?
- **Esfuerzo**: ¿Cuánto esfuerzo tuvo usted que hacer (mental y físicamente) para lograr su nivel de rendimiento?

Con estas dimensiones, a través de una escala ponderada, se obtuvo una visión completa de la carga de trabajo percibida por el usuario y fueron identificadas áreas que pueden generar sobrecarga cognitiva o emocional. El cuestionario se divide en dos fases. En una primera fase, los usuarios valoraron la tarea según las dimensiones mencionadas. Para cada una de ellas, cada usuario eligió un valor dentro del intervalo de 0 a 100 para reflejar su percepción sobre la aplicación. Los resultados se muestran en la Figura 5.8. Para medir

el nivel de carga de trabajo, se considera mayor carga de trabajo a calificaciones grandes o por encima de los 50 puntos y menor carga de trabajo a calificaciones menores a 50 puntos, lo cual sugiere que los usuarios pueden realizar sus tareas sin experimentar sobrecarga.

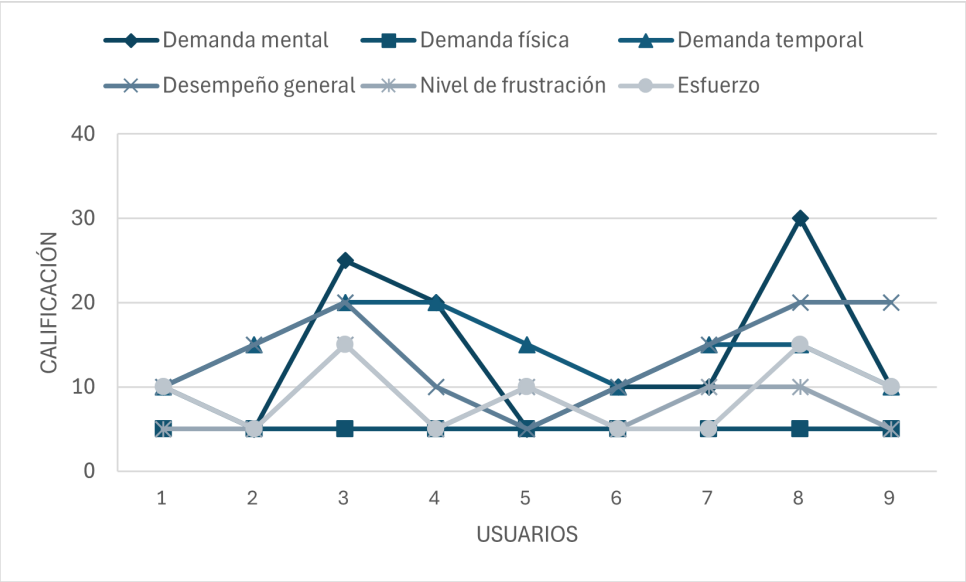


Figura 5.8: Resultados de las evaluaciones de las dimensiones.

En la segunda fase del cuestionario, cada usuario seleccionó entre pares de dimensiones, cuál, a su criterio, tenía mayor importancia. Con base a esto, se contó las veces que se repitió cada dimensión (peso) y se multiplicó por la evaluación que asignó el usuario en cada dimensión (*rating*). De este proceso resultó el *rating* ajustado (ver Ecuación 5.1). La Figura 5.9 muestra los resultados obtenidos de *rating* ajustado para cada dimensión.

$$Rating_{\text{ajustado}} = peso \times rating \tag{5.1}$$

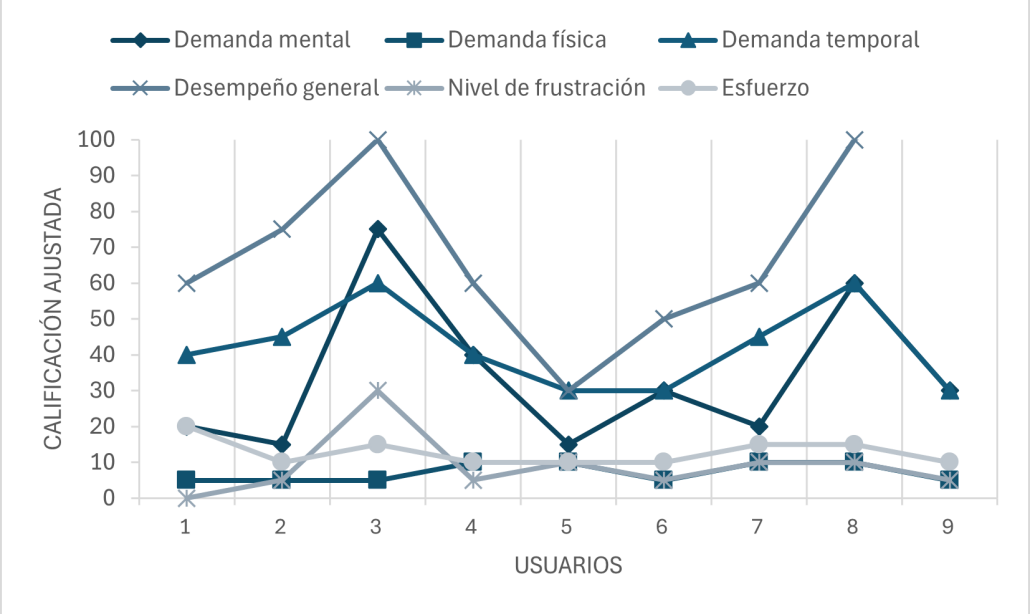


Figura 5.9: Resultados de las evaluaciones ajustadas en las dimensiones.

Luego se calcula el *rating* ponderado, que resulta de la suma de los *rating* ajustado de cada dimensión dividido entre el número de pares (ver Ecuación 5.2). La Figura 5.10 muestra los resultados obtenidos para el *rating* ponderado.

$$Rating_{\text{ponderado}} = \frac{Suma\ de\ ratings\ ajustados}{15} \tag{5.2}$$

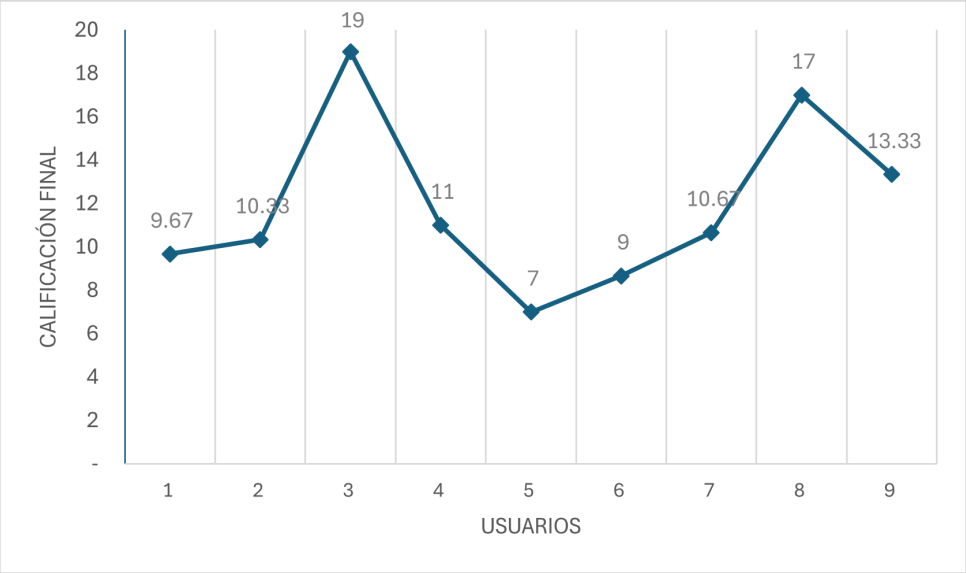


Figura 5.10: Resultados finales de las evaluaciones por usuario.

Los resultados obtenidos demuestran una calificación promedio para la carga de trabajo de $10{,}74 \pm 4{,}12$, que representan la media de los valores y la desviación estándar, teniendo un intervalo de confianza de [6.62, 14.86]. La Figura 5.11 muestra la calificación promedio resultante (10.74) y el intervalo de satisfacción y la carga de trabajo ([6.62, 14.86]) correspondiente. Esta calificación demuestra que el servicio de web semántica resultó de buena percepción y aceptación.

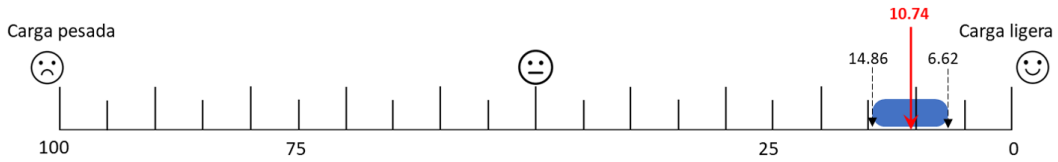


Figura 5.11: Resultados del cuestionario NASA-TLX para el servicio de web semántica.

5.3.2. Pruebas de percepción y usabilidad estética

Las pruebas para evaluar la percepción subjetiva del producto, se realizaron a través de la herramienta *AttrakDiff* [5]. Dicha herramienta es utilizada en estudios de usabilidad y diseño de experiencia de usuario, por lo que consideran tanto aspectos pragmáticos, que evalúan la funcionalidad y facilidad de uso, como hedónicos, que evalúan la respuesta emocional del usuario frente al diseño y su capacidad para generar placer y motivación durante el uso [3].

[5]Disponible en: *https://www.attrakdiff.de/index-en.html*

A través de un cuestionario estandarizado que proporciona la herramienta (ver Apéndice B), los usuarios evaluaron el producto utilizando pares de adjetivos opuestos (por ejemplo, simple - complicado, bueno - malo, original - ordinario, entre otros). Estas respuestas permitieron identificar cómo se percibe el producto en términos de usabilidad y atractivo emocional.

Los resultados de las pruebas se analizaron teniendo en cuenta cuatro dimensiones:

- **Atributos pragmáticos (PQ)**: evalúan la eficiencia y efectividad del producto para cumplir con los objetivos del usuario.
- **Atributos hedónicos de identificación (HQ-I)**: miden cómo el producto refleja la identidad del usuario y su capacidad de autoexpresión.
- **Atributos hedónicos de estimulación (HQ-S)**: evalúan la creatividad y estimulación que el producto ofrece al usuario.
- **Atractivo general (ATT)**: proporciona una visión global de la impresión general que el producto causa en el usuario.

Estas dimensiones se visualizaron en gráficos (portafolio de resultados, diagramas de valores medios y descripción de pares de palabras) que proporcionan una representación del rendimiento del producto, permitiendo un análisis detallado que facilita la toma de decisiones en el proceso de mejora del diseño. Las Figuras 5.12, 5.13 y 5.14, muestran los resultados obtenidos.

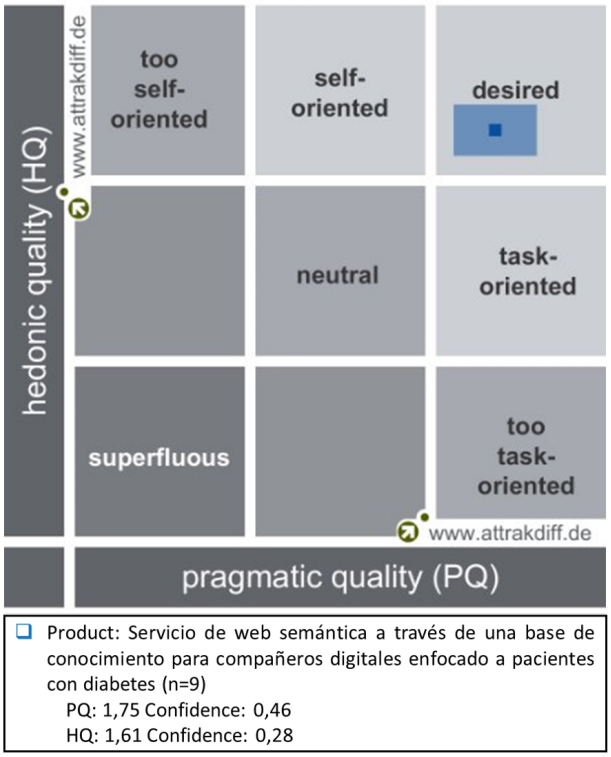


Figura 5.12: Portafolio de resultados.

En los resultados obtenidos se percibe que el producto es deseado por los usuarios, reflejando la calidad pragmática como la hedónica del producto, en cuanto a funcionalidad y atractivo emocional.

- El valor de PQ=1.75 indica que el producto es considerado funcional y cumple con las expectativas de los usuarios en cuanto a su facilidad de uso, con una confianza de 0,46. Esto sugiere una evaluación positiva.
- El valor de HQ=1.61 indica que el producto genera una experiencia emocional atractiva, lo que significa que es visto como estimulante e innovador por los usuarios, proporcionando placer y satisfacción más allá de su funcionalidad básica. La confianza de 0.28, en esta dimensión, sugiere que hubo consenso entre usuarios sobre las cualidades hedónicas del producto.
- El valor de HQ-I=1.54 sugiere que el producto también es visto como un producto que permite a los usuarios identificarse con él, lo que implica que los usuarios consideran que el producto es significativo y refleja sus necesidades y valores personales.
- El valor de HQ-S=1.68 denota que el producto genera estímulo e innovación, ofreciendo una experiencia atractiva que mantiene el interés y la curiosidad de los usuarios.
- El valor de ATT=2.29 indica que el atractivo general del producto es percibido como positivo. Este resultado combina las dimensiones pragmáticas y hedónicas, reflejando que los usuarios consideran el producto no solo útil, sino también agradable y satisfactorio en su conjunto.

De manera general, el análisis indica que el servicio de web semántica evaluado no sólo cumple con las expectativas funcionales de los usuarios, sino que también logra involucrar a los usuarios emocionalmente, ofreciéndoles una experiencia significativa y placentera. Estos resultados sugieren que el producto es deseado por los usuarios y su atractivo general refuerza el objetivo de ofrecer funcionalidad y satisfacción emocional.

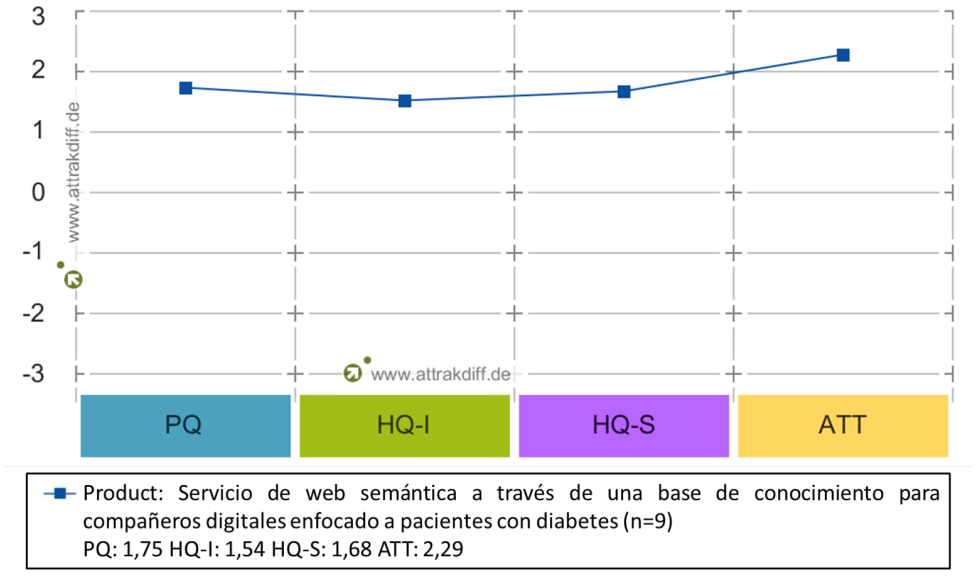


Figura 5.13: Diagrama de valores medios.

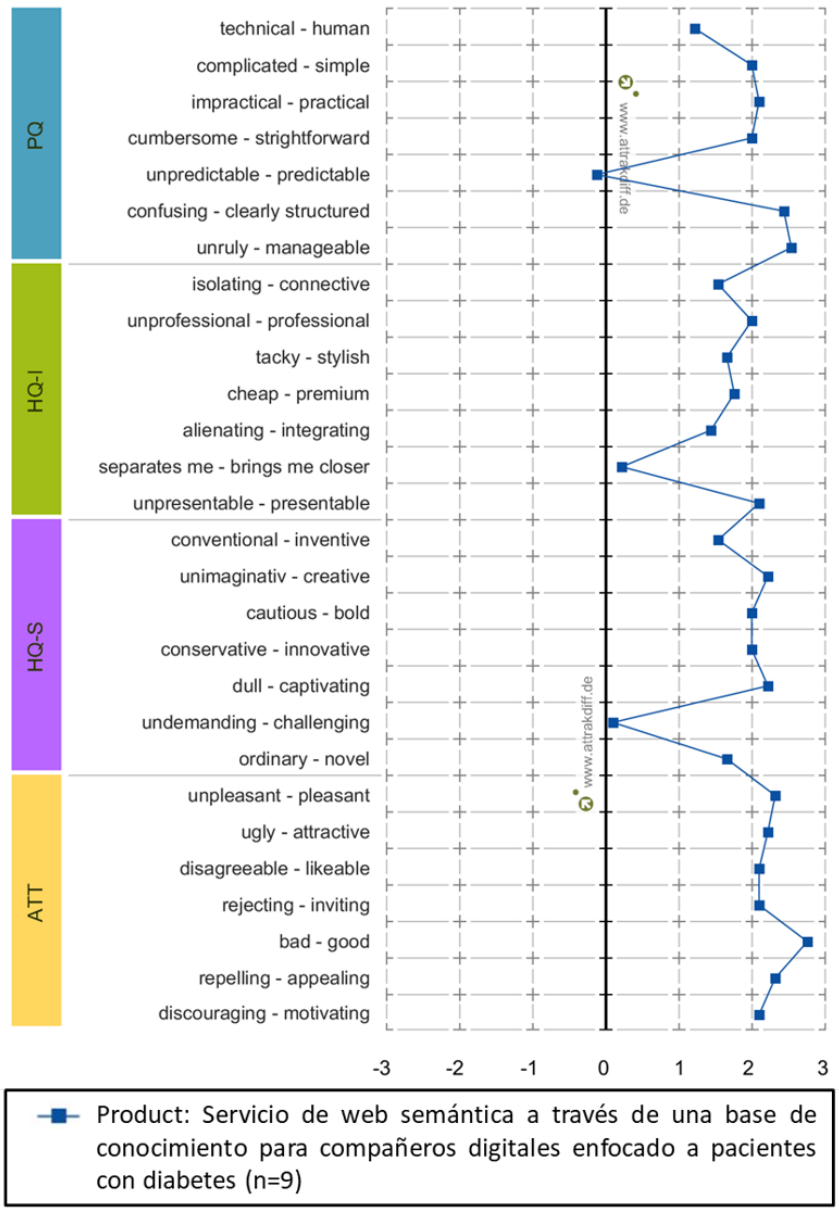


Figura 5.14: Descripción de pares de palabras.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Conclusiones generales

El estudio de diversos temas relacionados con las bases de conocimiento y servicios de web semántica permitió el desarrollo del presente trabajo de investigación, el cual arrojó como resultado un Servicio de Web Semántica a través de una Base de Conocimiento para compañeros digitales enfocado a pacientes con diabetes. De esta forma se le da cumplimiento al objetivo por el que fue desarrollado y a las tareas propuestas para su diseño e implementación. Es por ello que se puede llegar a las siguientes conclusiones:

- Se elaboró el marco teórico de la investigación sobre el estado del arte actual y trabajos relacionados con respecto a la propuesta realizada, teniendo en cuenta un conjunto de preguntas que sirvieron de ayuda para comparar las propuestas y detectar sus principales fortalezas y limitaciones.

- Se realizó el análisis y diseño para el desarrollo de los componentes, donde, primeramente, se analizaron los tópicos o requerimientos asociados a la diabetes. Como parte del proceso de análisis del mercado de datos `Diabetes`, fueron identificas las principales áreas de análisis, dimensiones y hechos. Se realizó la comparación de varias metodologías existentes para el desarrollo de este tipo de tecnología y se seleccionó la metodología de Kimball para guiar el proceso, así como la selección de las herramientas para el modelado de los datos.

- El proceso de análisis y diseño arrojó la identificación de tres tablas de hechos y nueve tablas de dimensiones, lo que generó el modelo de datos, el diseño de las transformaciones y la identificación de las clases, propiedades y relaciones de la ontología. También fue diseñada la arquitectura del servicio de web semántica, compuesta por cuatro componentes: el cliente, el servicio de web semántica RESTful, la base de conocimiento y el componente de búsqueda en la Web.

- La implementación del mercado de datos `Diabetes` permitió la definición de esquemas de almacenamiento, en los cuales están contenidas las tablas de hechos y dimensiones, y la integración de los datos, mediante la implementación de 12 transformaciones llevadas a cabo durante el proceso de ETL para la carga de los datos. La ontología desarrollada recopila y organiza el conocimiento relevante sobre la diabetes, cuenta con un total de 26 conceptos distribuidos entre clases y subclases y 10 propiedades o relaciones.

- El proceso de validación y pruebas de la ontología aseguró la integridad, consistencia y precisión de la misma. Este proceso abarcó la identificación y corrección de inconsistencias lógicas, la confirmación de la exactitud y completitud de los datos, y la evaluación de la eficiencia de las consultas y reglas de inferencia. Se utilizó

una combinación de herramientas automáticas y revisiones manuales para verificar que la ontología propuesta, no sólo es coherente y precisa, sino también capaz de proporcionar información valiosa y aplicable en contextos reales.

- La implementación del servicio de web semántica RESTful para la gestión de pacientes con diabetes, a través de una base de conocimiento, demuestra ser una solución robusta y escalable para la integración de diferentes fuentes de datos y la presentación de la información de manera accesible y amigable para los pacientes, ya que ofrece de forma automatizada, respuestas relacionadas con temas o conceptos importantes para los pacientes, como el tratamiento, la alimentación, el diagnóstico, etc.

- El uso combinado de una base de conocimiento y técnicas de Web *scraping* permite integrar datos de diversas fuentes. Esta dualidad garantiza que la información sobre la diabetes esté siempre actualizada y completa, incluso cuando algunos datos no estén disponibles en la base de conocimiento inicial. La flexibilidad del modelo, basado en RDF (*Resource Description Framework*), facilita la adición de nuevas fuentes de datos y la actualización del conocimiento, sin necesidad de cambios significativos en la estructura del sistema.

- La estructura modular del proyecto, con archivos dedicados para la configuración del servidor, las consultas SPARQL, la Web *scraping* y de la actualización de la base de conocimiento, promueve la mantenibilidad y escalabilidad del sistema. Además, el proceso automatizado de actualización de la base de conocimiento asegura que la información almacenada esté siempre alineada con las fuentes más recientes. Esta característica es esencial en el contexto de la gestión de datos médicos, donde la precisión y la actualidad de la información son críticas. Así mismo, la capacidad de realizar consultas SPARQL eficientes sobre la base de conocimiento permite una recuperación rápida y precisa de la información relevante. Este enfoque no sólo mejora el rendimiento del sistema, sino que también proporciona una base sólida para futuras extensiones y mejoras en las capacidades de búsqueda y análisis de datos.

- El correcto funcionamiento del mercado de datos `Diabetes` se verificó a través del Modelo V y con la utilización de la herramienta Casos de Pruebas; para el primero, se tuvo en cuenta su patrón de verificación y validación, con la realización de las pruebas unitarias, de integración y de sistema; y para el segundo, se definió un Caso de Prueba para las áreas de análisis: Tratamiento, Alimentación y Diagnóstico. Las pruebas aplicadas arrojaron un total de nueve No Conformidades distribuidas en los tres niveles de pruebas realizadas del modelo utilizado. Estas No Conformidades ya fueron resueltas, garantizando así, la calidad del producto final.

- Las pruebas a la aplicación se realizaron a través de los cuestionarios *NASA-TLX*, para medir la carga de trabajo y *AttrakDiff* para medir cómo los usuarios perciben el valor y el atractivo emocional del producto. Para ello, se tomó como muestra nueve usuarios, entre ellos, ocho pacientes con diabetes que varían en cuanto a tipo de diabetes, otras enfermedades que padecen y características de los pacientes, como edad, peso, altura, nivel de glucemia. También participó un médico que actuó como paciente, para validar las reglas establecidas sobre los tratamientos, el diagnóstico y la alimentación de un paciente.

- De las pruebas de carga de trabajo se obtuvo una clasificación promedio para la carga de trabajo de [10,74 ± 4,12], con un intervalo de confianza de [6.62, 14.86] y un promedio resultante de 10.74. Esto demuestra que el servicio de web semántica genera, en promedio, una carga ligera, lo que resulta ser aceptado y percibido positivamente por los usuarios.
- Las pruebas de experiencia de usuario también arrojaron un resultado aceptable, quedando el servicio de web semántica en el cuadrante de deseado por los usuarios. Esta conclusión se demuestra con un PQ=1.75, HQ-I=1.54, HQ-S=1.68 y ATT=2.29, lo que sugiere que el producto es atractivo y cumple con el objetivo de ofrecer satisfacción y funcionalidad a los usuarios.

6.2 Trabajo futuro

En el contexto actual de la salud digital, la gestión y el análisis de grandes volúmenes de datos, que apoyen el proceso de toma de decisiones, son esenciales para proporcionar un cuidado más personalizado y efectivo a los pacientes. Este enfoque permite la integración de datos clínicos y de otras fuentes relevantes, con el fin de ofrecer recomendaciones y monitoreo en tiempo real. Sin embargo, para asegurar la sostenibilidad y efectividad a largo plazo de este servicio, es crucial anticipar y planificar una serie de mejoras y expansiones. En particular, se identifican áreas clave para trabajos futuros, orientadas a garantizar la escalabilidad, la seguridad y la capacidad de adaptación del sistema.

Para impulsar la evolución del mercado de datos especializado en la diabetes, es esencial implementar una serie de iniciativas estratégicas orientadas tanto a optimizar su funcionamiento actual como a expandir su alcance. En particular, se propone la integración de otras enfermedades de alta prevalencia, como el cáncer y las enfermedades renales, con el objetivo de crear un ecosistema de datos más inclusivo y robusto. Esta ampliación no sólo enriquecería el valor del sistema existente, sino que también contribuiría al desarrollo de un repositorio integral de información sanitaria, adecuado para un espectro más amplio de condiciones médicas.

Un objetivo prioritario en este contexto radica en garantizar la escalabilidad del sistema. Este desafío implica no sólo la capacidad de gestionar volúmenes crecientes de datos, sino también la integración de otras Enfermedades No Transmisibles (ENT) en el mercado de datos. La inclusión de estas nuevas enfermedades requerirá ajustes y mejoras en la infraestructura tecnológica, a fin de asegurar que el sistema pueda expandirse y adaptarse sin comprometer su rendimiento ni la calidad de los datos.

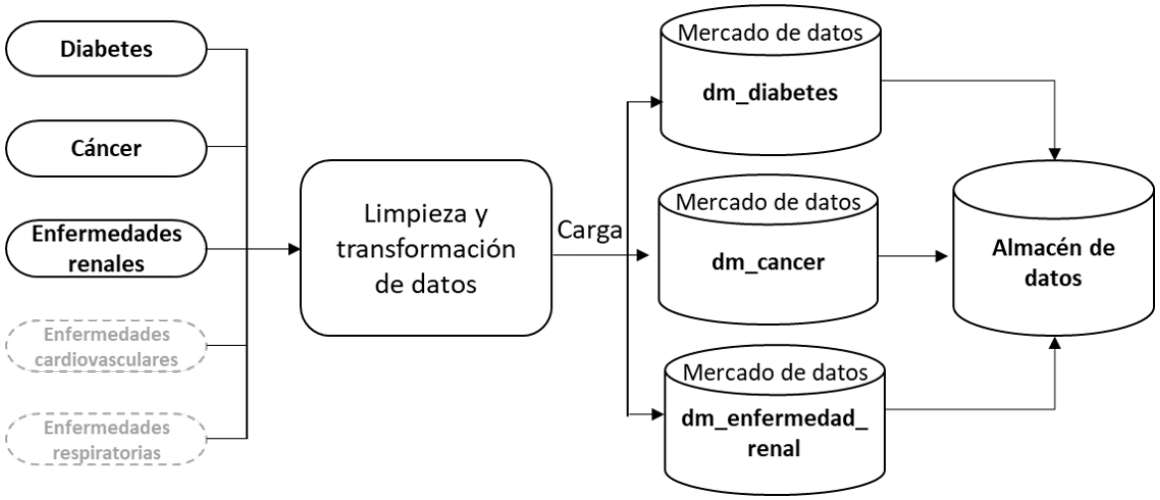


Figura 6.1: Inclusión de nuevas ENT.

Como se muestra en la Figura 6.1, la inclusión de otras ENT, supone la implementación de otros mercados de datos. Asimismo, se contempla la incorporación continua de nuevas fuentes de datos, a las que se les debe realizar un proceso de limpieza y transformación de los datos. Este proceso permitirá ampliar la base de conocimiento con la información de estas nuevas ENT y facilitar un entendimiento más detallado de las condiciones de los pacientes. Estas fuentes adicionales podrían incluir estudios clínicos, registros médicos electrónicos y dispositivos de monitoreo de salud, enriqueciendo de manera significativa la información disponible.

Otro aspecto fundamental en este ámbito consiste en la optimización de las técnicas de *scraping* de datos y la exploración del uso de APIs oficiales de sitios reconocidos. El *scraping* de datos, aunque crucial para la recolección de información, presenta desafíos en términos de eficiencia y seguridad. Por lo tanto, se prioriza la optimización de estas técnicas y, siempre que sea factible, se favorece el uso de APIs oficiales que ofrezcan acceso seguro y regulado a los datos, garantizando así la confiabilidad y legalidad de la información obtenida. La búsqueda de nuevos sitios con información relevante y certificada sobre temas de salud o específicamente de las ENT, amplía la veracidad en las respuestas a las solicitudes de los pacientes ante determinado tema y, a su vez, enriquece la base de conocimiento con la nueva información extraída.

La protección de los datos sensibles también ocupa un lugar destacado entre las prioridades. La implementación de medidas avanzadas de seguridad y privacidad es fundamental para asegurar la protección de la información de los pacientes, en estricto cumplimiento con las regulaciones vigentes en materia de protección de datos. Estas medidas incluyen la encriptación de datos, el control de acceso basado en roles y la auditoría regular de los sistemas de seguridad.

Adicionalmente, se debe establecer un plan de monitoreo y mantenimiento continuo del servicio de web semántica y la base de conocimiento. Este plan resulta fundamental para asegurar el correcto funcionamiento de estos componentes y su actualización frente a posibles cambios en las fuentes de datos o en las tecnologías empleadas. Un sistema de monitoreo proactivo permitirá detectar y resolver problemas, de manera anticipada, minimizando cualquier impacto en el servicio.

Por último, se propone la realización de pruebas exhaustivas y validaciones periódicas del servicio de web semántica y la base de conocimiento, con el propósito de garantizar su precisión, fiabilidad y capacidad de respuesta frente a diversos escenarios y volúmenes de

datos. La validación continua permitirá identificar y corregir posibles errores o desajustes, asegurando así que el sistema cumpla con los más altos estándares de calidad.

Apéndice A

Cuestionario NASA-TLX (*NASA-Task Load Index*)

NASA-TLX (*NASA-Task Load Index*) es un cuestionario diseñado para medir la carga de trabajo que percibe una persona al realizar determinada tarea. Su desarrollo data de los años 80 por la NASA, como parte del proceso para evaluar el esfuerzo mental, físico y temporal que experimentan los individuos, como pilotos o controladores aéreos, para realizar tareas complejas. De esta forma, el cuestionario es utilizado como herramienta estándar para medir la carga de trabajo en diversos contextos, ya sea en la industria, aplicaciones informáticas, entre otros [26].

El cuestionario se compone de seis dimensiones a evaluar:

- Demanda mental: evalúa el nivel de esfuerzo mental requerido por los individuos para realizar una tarea.
- Demanda física: se refiere al grado de esfuerzo físico necesario para completar una tarea.
- Demanda temporal: se refiere a la presión de tiempo o rapidez con la que determinada tarea debe ser completada.
- Desempeño general: evalúa la percepción del nivel de éxito en la ejecución de la tarea.
- Nivel de frustración: valora el nivel de estrés o frustración experimentado durante la realización de la tarea.
- Esfuerzo: mide cuánto trabajo percibió el individuo que debía realizar la tarea.

El proceso de ejecución del cuestionario tiene dos etapas:

1. Los participantes califican, en una escala de 0 a 100, cada una de las dimensiones.
2. Luego, realizan una comparación entre pares de dimensiones y ponderan la importancia de cada dimensión relativa a las demás.

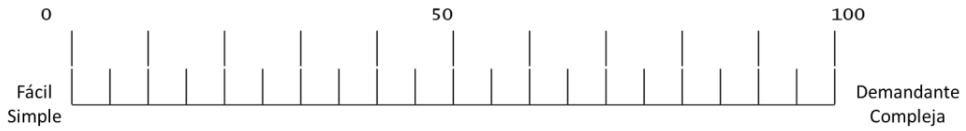
La puntuación final proporciona un índice que refleja la carga de trabajo experimentada por los participantes. De esta forma, se puede identificar qué aspectos de una tarea imponen más carga sobre los participantes.

Nombre:_____ Fecha: _____

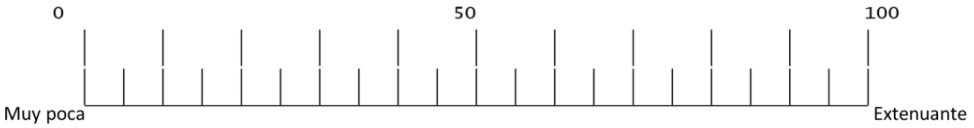
Escalas para la evaluación de la carga de trabajo

Para esta prueba se utilizará el índice de carga de tareas de la NASA (NASA-TLX) para evaluar la interfaz de usuario del servicio de web semántica.

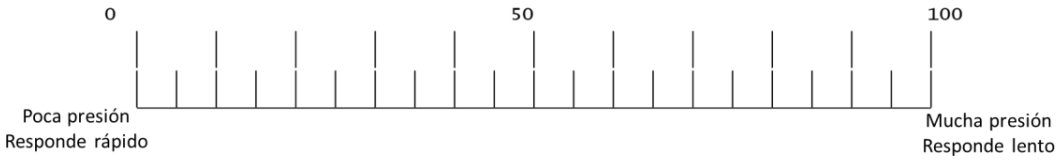
- **Demanda mental:**
  ¿Cuánta actividad mental y perceptiva requirió usted para comprender el funcionamiento de la aplicación? ¿La tarea fue fácil o exigente, simple o compleja?



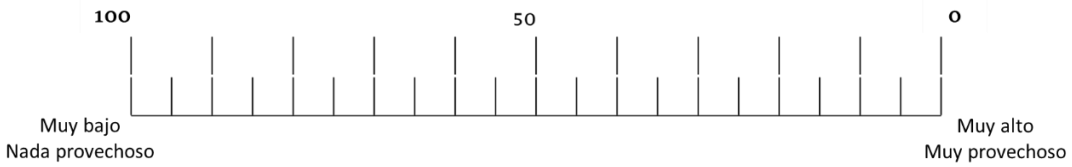
- **Demanda física:**
  ¿Cuánta actividad física necesitó usted para interactuar con la aplicación? ¿La tarea fue fácil o exigente, relajada o extenuante?



- **Demanda temporal:**
  ¿Cuánta presión de tiempo sintió usted debido al ritmo al que se procesaba una petición en la aplicación? ¿El ritmo era lento o rápido?

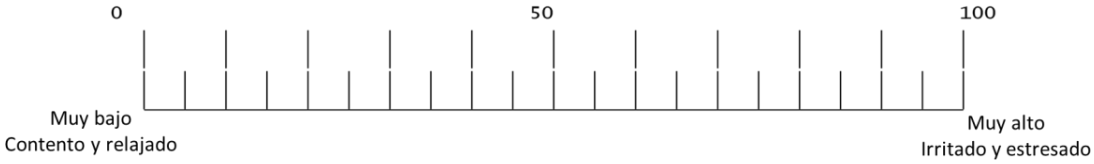


- **Desempeño general:**
  ¿Qué tan exitoso fue usted al realizar la tarea? ¿Qué tan satisfecho estuvo con su desempeño ?

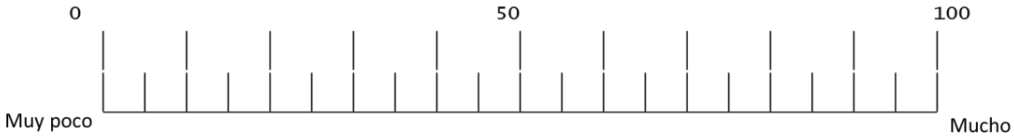


Nombre:_____ Fecha: _____

- **Nivel de frustración:**
  ¿Qué tan irritado, estresado y molesto versus contento, relajado y complaciente sintió usted durante la tarea?



- **Esfuerzo:**
  ¿Cuánto esfuerzo tuvo usted que hacer (mental y físicamente) para lograr su nivel de rendimiento?



Nombre:_____ Fecha: _____

Selección de importancia para las escalas de evaluación

Seleccione, de la siguiente tabla, la escala que le parezca más importante para medir la carga de trabajo de esta actividad. Debe elegir una escala en cada fila.

Demanda mental	Demanda física
Demanda mental	Demanda temporal
Demanda mental	Nivel de desempeño
Demanda mental	Nivel de frustración
Demanda mental	Esfuerzo
Demanda física	Demanda temporal
Demanda física	Nivel de desempeño
Demanda física	Nivel de frustración
Demanda física	Esfuerzo
Demanda temporal	Nivel de desempeño
Demanda temporal	Nivel de frustración
Demanda temporal	Esfuerzo
Nivel de desempeño	Nivel de frustración
Nivel de desempeño	Esfuerzo
Nivel de desempeño	Esfuerzo

Nombre:_____ Fecha: _____

Hoja de cálculo para obtener el índice de carga de trabajo para la aplicación

En la siguiente tabla, cuente el número de veces que eligió cada escala y coloque dicho número en la columna **Peso**. Coloque el *rating* (valoración) que asignó a cada escala en la columna ***Rating***. Calcule el ***rating* ajustado** para cada escala, sume estos resultados y divida entre 15 para obtener la evaluación final de carga de trabajo para esta tarea, i.e., el ***rating* ponderado**.

Nombre de la escala	**Peso**	***Rating***	***Rating* ajustado (Peso x *Rating*)**
Demanda mental			
Demanda física			
Demanda temporal			
Nivel de desempeño			
Nivel de frustración			

$$Suma\ de\ la\ columna\ Rating\ ajustado = \_\_\_\_\_$$

$$Rating\ ponderado = \frac{Suma\ de\ ratings\ ajustados}{15} = \_\_\_\_\_$$

Apéndice B

Cuestionario *AttrakDiff*

El cuestionario *AttrakDiff* es una herramienta que evalúa la experiencia de usuario, en la que se mide tanto aspectos pragmáticos como hedónicos de un producto. Ambas dimensiones ofrecen una visión integral de la usabilidad y el atractivo emocional que tiene el producto para los usuarios.

- Pragmático: se refiere a la usabilidad funcional del producto y mide la efectividad operativa del mismo. Evalúa la facilidad de uso del producto, eficiencia y la capacidad del mismo de ayudar a los usuarios a cumplir sus objetivos con el menor esfuerzo posible.
- Hedónico: se refiere a la capacidad del producto para generar una experiencia emocional positiva de manera que satisfaga las necesidades de los usuarios, más allá de lo funcional. Un producto con alta calidad hedónica, no solo resulta útil, sino también placentero y atractivo.

Las variables que mide el cuestionario *AttrakDiff* capturan los sentimientos subjetivos de los usuarios en relación con su interacción con el producto. Para ello, utiliza pares de adjetivos contrastantes para que los usuarios puedan valorar su experiencia.

Dimensiones que mide *AttrakDiff* [27]

1. Calidad pragmática (PQ): se refiere a la usabilidad y eficiencia funcional del producto. Determina cómo los usuarios interactúan con el producto de manera eficiente y sin dificultades para cumplir sus objetivos. Esta dimensión mide la capacidad del producto para satisfacer las necesidades operativas del usuario.
2. Calidad hedónica: mide el atractivo emocional y la experiencia subjetiva que los usuarios tienen con el producto. Se divide en dos subdimensiones:
   - Calidad hedónica de estimulación (HQ-S): esta subdimensión se enfoca en cómo el producto estimula al usuario, ya sea a nivel intelectual, emocional o sensorial. Un producto que puntúe alto en esta dimensión es catalogado como interesante, innovador y desafiante.
   - Calidad hedónica de identificación (HQ-I): esta subdimensión valora si el producto permite al usuario identificarse con el mismo, si se aprecia como algo que refleja su identidad personal, así como sus valores, gustos y preferencias.
3. Atractivo general (ATT): es una métrica global que integra tanto las percepciones pragmáticas como las hedónicas. Refleja la impresión global que el usuario tiene del producto. De esta manera, combina la funcionalidad del producto con el placer emocional que genera el mismo.

A continuación se muestra el cuestionario *AttrakDiff* aplicado al servicio de web semántica desarrollado.

Fecha: _____

Software a evaluar: Servicio de web semántica a través de una base de conocimiento para compañeros digitales enfocado a pacientes con diabetes.

Nombre: _____

Email: _____ Edad: _____

Nivel educativo: _____ Profesión: _____

Tiempo de experiencia usando servicios web: _____

Tiempo de experiencia usando la presente aplicación: _____

Evaluación de la percepción de usabilidad y estética

Marque un círculo de acuerdo con su percepción del producto respecto al par de las palabras en los extremos. No piense demasiado en las palabras, elija la opción de manera espontánea. Aun así, siente que el par de palabras no aplica al producto, elija una opción de igual manera. Recuerde que no hay calificaciones "correcta o incorrecta", su opinión es lo que cuenta.

Para cada para de palabras, marque el círculo que considere una descripción más apropiada del producto.

Humano	○	○	○	○	○	○	○	Técnico
Aislante	○	○	○	○	○	○	○	Conectivo
Agradable	○	○	○	○	○	○	○	Desagradable
Inventivo	○	○	○	○	○	○	○	Convencional
Simple	○	○	○	○	○	○	○	Complicado
Profesional	○	○	○	○	○	○	○	No profesional
Feo	○	○	○	○	○	○	○	Atractivo
Práctico	○	○	○	○	○	○	○	Impráctico
Agradable	○	○	○	○	○	○	○	Desagradable
Pesado de usar	○	○	○	○	○	○	○	Sencillo de usar

Elegante	◯	◯	◯	◯	◯	◯	◯	Cursi
Predecible	◯	◯	◯	◯	◯	◯	◯	Impredecible
Barato	◯	◯	◯	◯	◯	◯	◯	De calidad
Alienante	◯	◯	◯	◯	◯	◯	◯	Integrante
Me acerca a la gente	◯	◯	◯	◯	◯	◯	◯	Me separa de la gente
Impresentable	◯	◯	◯	◯	◯	◯	◯	Presentable
Repelente	◯	◯	◯	◯	◯	◯	◯	Provocativo
No imaginativo	◯	◯	◯	◯	◯	◯	◯	Creativo
Bueno	◯	◯	◯	◯	◯	◯	◯	Malo

Confuso	◯	◯	◯	◯	◯	◯	◯	Claramente estructurado
Repelente	◯	◯	◯	◯	◯	◯	◯	Atractivo
Valiente	◯	◯	◯	◯	◯	◯	◯	Cauteloso
Innovador	◯	◯	◯	◯	◯	◯	◯	Conservador
Aburrido	◯	◯	◯	◯	◯	◯	◯	Cautivante
Poco exigente	◯	◯	◯	◯	◯	◯	◯	Desafiante
Motivante	◯	◯	◯	◯	◯	◯	◯	Desalentador
Original	◯	◯	◯	◯	◯	◯	◯	Ordinario
Incidentado	◯	◯	◯	◯	◯	◯	◯	Manejable

Apéndice C

Opiniones de los usuarios

En el contexto del servicio de web semántica implementado en el presente trabajo de investigación, los pacientes han compartido sus opiniones:

Opinión # 1:

> «Este servicio ha sido un cambio radical en la forma en que manejo mi diabetes. La información sobre alimentación y el tratamiento personalizado me han ayudado a tomar decisiones más informadas sobre mi salud.»
> — Paciente # 1

Opinión # 2:

> «La posibilidad de recibir recomendaciones basadas en mis niveles de glucosa es increíble. Ahora puedo ajustar mi dieta y tratamiento de manera más efectiva. Me gustaría que incluyeran más recetas específicas y opciones de comida que se adapten a mis gustos.»
> — Paciente # 2

Opinión # 3:

> «El servicio me proporciona un sentido de control que antes no tenía. Entender cómo mis decisiones afectan mis niveles de glucosa es clave para mí. Sería beneficioso tener acceso a un chat en línea con expertos en diabetes para resolver dudas rápidamente.»
> — Paciente # 3

Opinión # 3:

> «Como médico, he observado cómo la gestión de la diabetes puede ser abrumadora para muchos pacientes. El Servicio de Web Semántica representa un avance significativo en el apoyo que brindamos a los pacientes. Al proporcionar información personalizada sobre el tratamiento y la alimentación con base en los datos clínicos de los pacientes y a sus preferencias individuales, este sistema no sólo mejora el control de la diabetes, sino que también empodera a los pacientes para tomar decisiones informadas sobre su salud.»
> — Médico

Índice de figuras

Índice de tablas

Índice de códigos

Bibliografía

[1] National Aeronautics y Space Administration. *NASA TLX-Task Load Index*. 2020. URL: https://humansystems.arc.nasa.gov/groups/TLX/ (visitado 06-09-2024).

[2] American Diabetes Association. *Diabetes*. 1995-2024. URL: https://diabetes.org/ (visitado 06-07-2024).

[3] AttrakDiff. *Your benefits*. URL: https://www.attrakdiff.de/index-en.html (visitado 06-09-2024).

[4] Yasser Azán Basallo, Anay Díaz Estrada y Salvador González Gómez. «Una experiencia en integración de aplicaciones empresariales». En: *Revista Cubana de Ciencias Informáticas* 3.3-4 (2009), págs. 13-18. ISSN: 1994-1536. URL: https://www.redalyc.org/pdf/3783/378343637001.pdf.

[5] S. Bansal et al. «Generalized semantic Web service composition». En: *Service Oriented Computing and Applications* 10.2 (2016), págs. 111-133. DOI: 10.1007/s11761-014-0167-5.

[6] D. R. Bernabeu. *Hefesto: Metodología para la construcción de un Data Warehouse*. 2010.

[7] BioPortal. *BioMedBridges Diabetes Ontology*. 2015. URL: https://bioportal.bioontology.org/ontologies/DIAB (visitado 05-03-2024).

[8] BioPortal. *Chinese Diabetes Mellitus Ontology*. 2023. URL: https://bioportal.bioontology.org/ontologies/CDO (visitado 05-03-2024).

[9] BioPortal. *Diabetes Mellitus Diagnosis and Support Ontology*. 2022. URL: https://bioportal.bioontology.org/ontologies/DMDSONT (visitado 05-03-2024).

[10] BioPortal. *Ontology of Glucose Metabolism Disorder*. 2021. URL: https://bioportal.bioontology.org/ontologies/OGMD (visitado 05-03-2024).

[11] María E. Chávez, Oscar Cárdenas y Oscar Benito. «La web semántica». En: *Revista de investigación de Sistemas e Informática)*. Vol. 2. 3. 2005, págs. 43-54.

[12] DATA y KNOWLEDGE GROUP. *HermiT OWL Reasoner, The New Kid on the OWL Block*. 2021. URL: http://www.hermit-reasoner.com/ (visitado 14-06-2024).

[13] Anthony Debons. «Foundations of Information Science». En: ed. por Marshall C. Yovits. Vol. 31. Advances in Computers. Elsevier, 1990, págs. 325-378. DOI: https://doi.org/10.1016/S0065-2458(08)60156-4. URL: https://www.sciencedirect.com/science/article/pii/S0065245808601564.

[14] API DeepL. *DeepL API Docs*. 2024. URL: https://developers.deepl.com/docs/v/es (visitado 06-07-2024).

[15] Omkar Deshpande et al. «Building, maintaining, and using knowledge bases: a report from the trenches». En: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 2013, págs. 1209-1220.

[16] T. H. Devenport y L. Prusak. *Working Knowledge: how organisations manage what they know*. Harvard University Press, 1998.

[17] Zaidi Fayçal y Tari Abdelkamel. «Building a semantic web services ontology in the pharmaceutical field using the OWL-S Language». En: *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*. 2021, págs. 1-8. DOI: 10.1109/ICISAT54145.2021.9678408.

[18] Flask. *Flask*. 2019. URL: https://flask.palletsprojects.com/en/3.0.x/ (visitado 18-05-2024).

[19] The Apache Software Foundation. *Apache Jena*. 2024. URL: https://jena.apache.org/index.html (visitado 15-06-2024).

[20] Kimberly García et al. «Proactive Digital Companions in Pervasive Hypermedia Environments». En: *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*. 2020, págs. 54-59. DOI: 10.1109/CIC50333.2020.00017.

[21] John H Gennari et al. «The evolution of Protégé: an environment for knowledge-based systems development». En: *International Journal of Human-Computer Studies* 58.1 (2003), págs. 89-123. ISSN: 1071-5819. DOI: https://doi.org/10.1016/S1071-5819(02)00127-1. URL: https://www.sciencedirect.com/science/article/pii/S1071581902001271.

[22] Fabio Giachelle et al. «Searching for Reliable Facts over a Medical Knowledge Base». En: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '23. Taipei, Taiwan: Association for Computing Machinery, 2023, págs. 3205-3209. ISBN: 9781450394086. DOI: 10.1145/3539618.3591822. URL: https://doi.org/10.1145/3539618.3591822.

[23] Gobierno de México, Secretaría de Salud, Comisión Coordinadora de Institutos Nacionales de Salud y Hospitales de Alta Especialidad. *Diabetes en México*. 2023. URL: https://www.gob.mx/promosalud/acciones-y-programas/diabetes-en-mexico-284509.

[24] Liliana González Palacio. «MÉTODO PARA GENERAR CASOS DE PRUEBA FUNCIONAL EN EL DESARROLLO DE SOFTWARE». Español. En: *Revista Ingenierías Universidad de Medellín* (2009). ISSN: 1692-3324. URL: https://www.redalyc.org/articulo.oa?id=75017199005.

[25] Md. Gulzar y Muqeem Ahmed. «Chronic Disease Management using Semantic Web Technologies». En: *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2023, págs. 1629-1633.

[26] Sandra G. Hart y Lowell E. Staveland. «Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research». En: *Human Mental Workload*. Ed. por Peter A. Hancock y Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, págs. 139-183. DOI: 10.1016/S0166-4115(08)62386-9. URL: https://www.sciencedirect.com/science/article/pii/S0166411508623869.

[27] Marc Hassenzahl, Franz Koller y Michael Burmester. *Der User Experience (UX) auf der Spur: Zum Einsatz von www.attrakdiff.de*. Tagungsband UP08. Stuttgart, 2008.

[28] Agency for Healthcare Research y Quality. *NASA Task Load Index*. URL: https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index (visitado 06-09-2024).

[29] IBM. *Visión general de servicios Web*. 2021. URL: https://www.ibm.com/docs/es/rsas/7.5.0?topic=applications-web-services-overview (visitado 21-02-2024).

[30] W.H. Inmon. *Building the Data Warehouse*. Wiley, 2005. ISBN: 9780471774235. URL: https://books.google.com.mx/books?id=QFKTmh5IFS4C.

[31] R. Kimball y J. Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley, 2011. ISBN: 9781118079683. URL: https://books.google.com.mx/books?id=TCLfzU2ilVkC.

[32] R. Kimball et al. *The Data Warehouse Lifecycle Toolkit, 2nd Ed*. Wiley India Pvt. Limited, 2008. ISBN: 9788126516896. URL: https://books.google.com.mx/books?id=4EV5lzDhjNoC.

[33] Montoya J. L. *Pentaho Data Integration en acción*. 2019. URL: https://openwebinars.net/blog/que-es-y-como-usar-pentaho-data-integration-tutorial-en-espanol/ (visitado 18-05-2024).

[34] R. Labrador. «Base de Conocimientos». En: (2018).

[35] Manfred Langen y Sabrina Heinrich. «Humanoid Robots: Use Cases as AI-Lab Companion : Can an empathic and collaborative digital companion motivate innovation?» En: *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. 2019, págs. 1-6. DOI: 10.1109/ICE.2019.8792614.

[36] Chin-Feng Lee, Tz-Ruei Kang y Hsing-Yu Hou. «Investigating Factors Influencing the Retention Intention of Digital Companion for Learning Project: Case of a Technology University in Taiwan». En: *2023 12th International Conference on Awareness Science and Technology (iCAST)*. 2023, págs. 321-325. DOI: 10.1109/iCAST57874.2023.10359317.

[37] Anthony Liew. «DIKIW: Data, Information, Knowledge, Intelligence, Wisdom and their Interrelationships». En: *Business Management Dynamics* (abr. de 2013).

[38] Juan Antonio Lossio-Ventura et al. «OC-2-KB: A software pipeline to build an evidence-based obesity and cancer knowledge base». En: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2017, págs. 1284-1287. DOI: 10.1109/BIBM.2017.8217845.

[39] Ricardo Alonso Maturana. *Pellet OWL Reasoner*. 2009. URL: https://nextweb.gnoss.com/recurso/pellet-owl-reasoner/4879d59f-38e0-42f8-aa59-12b8288cd6da (visitado 04-06-2024).

[40] Mohit Mayank. «A guide to Knowledge Graphs: A consolidation of notes that briefly but gently introduces Knowledge graphs and shines a light on several practical aspects». En: *Towards Data Science* (2021).

[41] Organización Panamericana de la Salud. *Diabetes*. 2023. URL: https://www.paho.org/es/temas/diabetes.

[42] Levent V. Orman. «Knowledge base architecture». En: *Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems*. SIGBDP '90. Orlando, Florida, USA: Association for Computing Machinery, 1990, págs. 325-336. ISBN: 0897914163. DOI: 10.1145/97709.97732. URL: https://doi.org/10.1145/97709.97732.

[43] pgAdmin. *pgAdmin*. 2024. URL: https://www.pgadmin.org/ (visitado 18-05-2024).

[44] Zhu Ping y Tan Yuanhua. «Digitizing construction method of large knowledge base system». En: *2015 International Conference on Computer and Computational Sciences (ICCCS)*. 2015, págs. 279-283. DOI: 10.1109/ICCACS.2015.7361366.

[45] PostgreSQL. *The World's Most Advanced Open Source Relational Database*. 2024. URL: https://www.postgresql.org/ (visitado 18-05-2024).

[46] *PROYECTO de Norma Oficial Mexicana PROY-NOM-015-SSA2-2018, Para la prevención, detección, diagnóstico, tratamiento y control de la Diabetes Mellitus*. 2018. URL: https://www.dof.gob.mx/nota_detalle.php?codigo=5521405&fecha=03/05/2018#gsc.tab=0 (visitado 06-07-2024).

[47] Ma Qiang, Xu Tao y Daiyu Gang. «Research and implementation of Archives knowledge Base for Multi-source heterogeneous data Fusion». En: *2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. 2022, págs. 462-465. DOI: 10.1109/ICFTIC57696.2022.10075283.

[48] Iara Margolis Ribeiro y Bernardo Providencia. «Quality perception with Attrakdiff method: a study in higher education». En: *Advances in Design and Digital Communication: Proceedings of the 4th International Conference on Design and Digital Communication, Digicom 2020, November 5–7, 2020, Barcelos, Portugal*. Springer. 2021, págs. 222-233.

[49] Sergey Rippa y Nataliia Kasatkina. «Knowledge base as informatization project». En: *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*. Vol. 02. 2013, págs. 583-586. DOI: 10.1109/IDAACS.2013.6662991.

[50] Rudi Studer, V. Richard Benjamins y Dieter Fensel. «Ingeniería del conocimiento: principios y métodos». En: *Ingeniería de datos y conocimiento* (1998), págs. 161-197.

[51] Jeff Sucari. *Modelo v*. URL: https://www.academia.edu/38757252/Modelo_v (visitado 06-07-2024).

[52] Muyao Tang et al. «Intelligent Dental Triage System Oriented on Dental Symptom Knowledge Base». En: *Proceedings of the 3rd International Symposium on Artificial Intelligence for Medicine Sciences*. ISAIMS '22. Amsterdam, Netherlands: Association for Computing Machinery, 2022, págs. 452-456. ISBN: 9781450398442. DOI: 10.1145/3570773.3570793. URL: https://doi.org/10.1145/3570773.3570793.

[53] Google Cloud Translate. *API Cloud Translation*. 2024. URL: https://cloud.google.com/translate/?hl=es (visitado 06-07-2024).

[54] Jair A. Villanueva y Fabiola M. Martinez. «Toward a Health Care Technology Management Knowledge Base». En: *2009 Pan American Health Care Exchanges*. 2009, págs. 127-129. DOI: 10.1109/PAHCE.2009.5158381.

[55] Huaqiong Wang, Xiaoyu Miao y Pan Yang. «Design and Implementation of Personal Health Record Systems Based on Knowledge Graph». En: *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*. 2018, págs. 133-136. DOI: 10.1109/ITME.2018.00039.

[56] Patricia Layzell Ward. *Harrod's Librarians' Glossary and Reference Book*. Vol. 21. 8. Emerald Group Publishing Limited, 2000, págs. 443-447.

[57] Carmen Gloria Wolff. *Implementando un DataWarehouse*. 5. Departamento de Ingeniería Informática y Ciencias de la Computación, 2000, pág. 4.

[58] Qiong Wu. «Construction of English Grammar Knowledge Base System Based on DB2 Database». En: *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*. 2022, págs. 1-5. DOI: 10.1109/ICKECS56523.2022.10059809.

[59] Alexander Wurl et al. «A Conceptual Design of a Digital Companion for Failure Analysis in Rail Automation». En: *2019 IEEE 21st Conference on Business Informatics (CBI)*. Vol. 01. 2019, págs. 578-583. DOI: 10.1109/CBI.2019.00073.

[60] Suna Yin, Dehua Chen y Jiajin Le. «Deep Neural Network Based on Translation Model for Diabetes Knowledge Graph». En: *2017 Fifth International Conference on Advanced Cloud and Big Data (CBD)*. 2017, págs. 318-323. DOI: 10.1109/CBD.2017.62.

[61] Chi Zheng et al. «A Chronic Disease Self-Management System Based on OWL-Based Ontologies and Semantic Rules». En: *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*. 2016, págs. 1-6. DOI: 10.1109/ITME.2016.0011.