PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UN NAVEGADOR DE CONCEPTOS ENLAZADOS EN EL DOMINIO DE CIENCIAS DE LA COMPUTACIÓN

Tesis para optar por el **Título de Ingeniero Informático**, que presenta el bachiller:

Alexis Enrique León Shimabukuro

Asesor: Héctor Andrés Melgar Sasieta

Lima, julio del 2018

A mis padres, por el apoyo incondicional

que siempre me brindaron

A mis amigos, por las gratas experiencias

vividas a lo largo de la carrera

Agradecimientos

A mi asesor, por su apoyo y orientación

brindados en el desarrollo de mi tesis

A todos los profesores que

contribuyeron a mi formación académica

Resumen

En la actualidad, la World Wide Web es una de las fuentes principales de información, siendo un espacio que se encuentra en constante crecimiento, puesto que cada vez mas personas cuentan con acceso a internet. Sin embargo, esto genera múltiples problemas entre los que podemos mencionar como la duplicidad de información, que dificulta la búsqueda de información relevante a los usuarios, quienes utilizan herramientas como motores de búsqueda para esta tarea.

Ante esta situación surgió la Web Semántica, extensión de la Web tradicional, en donde la información es comprensible tanto para las personas como para las máquinas. Para publicar información en este espacio existen un conjunto de prácticas conocido como *Linked Data*, que permiten que la información se estructure según su significado y relación entre los datos que la componen, lo que facilita la labor de búsqueda y permite el descubrimiento de nueva información, generando valor a usuarios como investigadores, que constantemente se encuentran en búsqueda de conocimientos.

Toda la información en constante crecimiento contenida en la Web Semántica puede ser accedida simplemente mediante navegadores convencionales; sin embargo, esta se encuentra en su mayoría en formato *RDF*, por lo que el usuario común no podrá comprender su contenido. Para que la información pueda ser de utilidad, se necesitan conocimientos en conceptos como *RDF* y *XML*, lo que limita gran parte del potencial actual de la Web Semántica a los especialistas en dicha área.

El presente proyecto implementa un navegador de *Linked Data*, mediante el cual los usuarios pueden consultar información en el dominio de las ciencias de la computación, dicha información es obtenida de la Web Semántica, permitiendo el descubrimiento de información relevante, contribuyendo así a la expansión de dicha tecnología, que busca unificar y estructurar toda la información contenida en la web.

Para la elaboración del proyecto, se implementó un módulo de procesamiento de consultas, en donde el usuario ingresa una cadena de búsqueda, al igual que en un motor de búsqueda tradicional y mediante esta cadena se obtienen posibles propiedades, que

son enviadas a manera de consultas en lenguaje *SPARQL*, a partir de cuyos resultados se construyen estructuras *RDFs* que muestran los conceptos y la información presentada en una interfaz gráfica para que el usuario pueda visualizarla y navegar a través de dichos conceptos, permitiendo el descubrimiento de información relevante.

Tabla de contenido

Índice de ilustraciones

Índice de tablas

Capítulo 1. DEFINICIÓN DEL PROBLEMA

1.1. Problemática

Desde su creación, la World Wide Web ha contribuido a mejorar el acceso a la información, eliminando las barreras físicas para compartir documentos a nivel global. Para esto dispone de aplicaciones como navegadores web, que permiten a los usuarios acceder a este gran espacio y motores de búsqueda, que procesan los documentos contenidos en el mismo para presentar información relevante y satisfacer sus necesidades (Bizer, Heath, & Berners-Lee, 2009).

Inicialmente la Web, al ser diseñada como un espacio de información, tenía como objetivo no solo facilitar la comunicación entre personas, sino también la de las máquinas, sin embargo, debido al exorbitante crecimiento de esta, la información en la web ha sido estructurada para ser entendible de manera sencilla para los usuarios. Los documentos en la web se encuentran en formato *HTML (HyperText Markup Language*), un lenguaje de marcado que se enfoca en la estructura de la página, mas no en el contenido de la información de esta. Cada vez más personas tienen acceso a internet; en 1995, aproximadamente 35 millones de personas era el total del grupo que tenía acceso a la red, comparado a los 2.8 mil millones de usuarios registrados para el 2014 (Meeker, 2015). Esto contribuyó al crecimiento desordenado de la web, la información se encuentra dispersa en este gran espacio, en donde los motores de búsqueda son utilizados para brindar al usuario la información deseada, dentro de sitios webs posiblemente asociados al dominio de la búsqueda (Berners-Lee, Hendler, & Lassila, 2001).

Dentro de este contexto, al existir tanta información contenida en documentos, es inevitable encontrarse con información duplicada o redundante, la duplicidad puede ser completa o parcial, en el mismo lenguaje o incluso uno diferente; esto genera una pérdida de tiempo al querer obtener información sobre un tema en particular; el usuario deberá navegar en la web sobre un gran número de documentos, encontrándose con información duplicada, por ejemplo, en el ámbito científico, publicaciones redundantes pueden contener los mismos

resultados, sin aportar información adicional (Alfonso, Bermejo, & Segovia, 2005).

Por otra parte, si el usuario no es un especialista en el dominio de la información a investigar, la búsqueda se dificultaría aún más. La información se encuentra dispersa a lo largo de todo el espacio que conforma la web; los motores de búsqueda reducen la tarea de búsqueda considerablemente, pero la eficiencia con la que devuelven resultados se basa en la precisión de la cadena de búsqueda. De esta manera la estructura de la web actual dificulta el descubrimiento de nueva información.

Sin embargo, con el paso de los últimos años, la web ha ido evolucionando de ser sólo un gran repositorio de documentos a un espacio que conecte dichos documentos con la data contenida en ellos, para esto se desarrollaron un conjunto de prácticas para estructurar y publicar la información conocidas como *Linked Data*. Mediante esta evolución, la web actual o "Web de documentos" se extendería para convertirse en una "Web de datos", también denominada Web Semántica, en donde las máquinas podrán procesar la información en base al contenido de esta (Berners-Lee et al., 2001).

Para lograr esto la información se estructura mediante un esquema que permita la interoperabilidad entre los sistemas actuales, este esquema se denomina *RDF (Resource Description Framework)*, que permite modelar y describir recursos, asignándoles propiedades acordes al dominio de conocimiento correspondiente ("RDF - Semantic Web Standards," n.d.). Para lograr dicho nivel de especificación se utilizan ontologías como modelo, dado que permiten representar el conocimiento y el descubrimiento de nuevas relaciones. Los recursos representados por este esquema son implementados mediante URIs, cadenas de caracteres que identifican recursos únicos. Para consultar dichos recursos en la Web Semántica se utiliza el lenguaje *SPARQL* ("SPARQL Query Language for RDF," n.d.).

Toda la información en constante crecimiento contenida en la Web Semántica puede ser accedida simplemente mediante navegadores convencionales; sin

embargo, esta se encuentra en su mayoría en formato *RDF*, por lo que el usuario común no podrá comprender su contenido. Para que la información pueda ser de utilidad, se necesitan conocimientos en conceptos como *RDF* y *XML*, lo que limita gran parte del potencial actual de la Web Semántica a los especialistas en dicha área. Actualmente, dicha información se encuentra en repositorios que contienen datasets en formato *RDF*, algunos de los principales repositorios son *DBpedia*, que contiene la información de Wikipedia en formato RDF, dicho repositorio cuenta con un poco más de 103 millones de tripletas *RDF*, en donde se han estructurado algunos conceptos incluso entre diferentes dominios. Otro gran repositorio es el del proyecto Bio2RDF, que contiene conocimiento del genoma humano y medicina, contando con 2.3 billones de tripletas RDF ("TaskForces/CommunityProjects/LinkingOpenData/DataSets," n.d.).

Ante esta situación, se propone implementar un navegador de *Linked Data*, mediante el cual los usuarios podrán consultar información en el dominio de las ciencias de la computación, dicha información provendrá de la Web Semántica, permitiendo el descubrimiento de información relevante, contribuyendo así a la expansión de dicha tecnología, que busca unificar y estructurar toda la información contenida en la web. Es importante mencionar que no se necesitará conocimiento previo en tecnologías asociadas como *RDF*; esta herramienta dispondrá de los servicios para satisfacer la necesidad del usuario.

1.2. Árbol de Problemas

Se presenta el árbol de problemas que resume la problemática planteada. En el nivel más bajo se presentan las causas, en el nivel medio, el problema y, en el nivel más alto, los efectos que genera este problema.

ARBOL DE PROBLEMAS		1	2	3	4
	EFECTOS	Gran esfuerzo requerido para la búsqueda de información relevante.	Limitación del acceso al contenido de la información de la Web Semántica	Información obtenida que no satisface la necesidad del usuario.	
	PROBLEMA	Dificultad para obtener información relevante en la Web Semántica			
	CAUSAS	Desconocimiento de la información requerida por parte del usuario.	Desconocimiento de las herramientas provistas por la Web Semántica.	Dificultad para acceder a la información de la nube de Linked Data.	Falta de herramientas que permitan obtener información de la Web semántica.

Ilustración 1. Árbol de problemas

1.3. Objetivo general

Diseñar e implementar un navegador de conceptos enlazados en el dominio de ciencias de la computación.

1.4. Objetivos específicos

OE1. Diseñar e implementar un mecanismo de transformación de una consulta dentro del dominio de ciencias de la computación.

OE2. Diseñar e implementar un mecanismo que permita obtener y procesar información de la nube de Linked Open Data.

OE3. Diseñar e implementar un mecanismo que permita la extracción de información y navegación dentro una estructura RDF.

OE4. Diseñar e implementar un componente que permita la visualización de la información.

1.5. Resultados esperados

R1. Módulo de procesamiento de lenguaje natural para procesar la consulta. (OE1)

R2. Estructura RDF asociada al recurso obtenido. (OE2)

R3. Mecanismo que permita obtener información relacionada entre trabajos académicos. (OE3)

R4. Estructura RDF desreferenciable que permita la navegación entre conceptos. (OE3)

R5. Navegador con una interfaz gráfica intuitiva. (OE4)

1.6. Herramientas, métodos, metodologías y procedimientos

1.6.1. Introducción

En esta sección se presentarán todas las herramientas, métodos, metodologías y procedimientos a utilizar para el desarrollo de los resultados esperados, describiendo y justificando cada uno.

Resultado esperado	**Herramienta utilizada**
Módulo de procesamiento de lenguaje natural para procesar la consulta.	Eclipse Java
Estructura RDF asociada al recurso obtenido.	Apache Jena SPARQL
Mecanismo que permita obtener información relacionada entre trabajos académicos	Apache Jena SPARQL OWL: Web Ontology Language
Estructura RDF desreferenciable que permita la navegación entre conceptos.	Apache Jena OWL: Web Ontology Language

	SPARQL
Navegador con una interfaz gráfica intuitiva.	Eclipse Java Spring

Tabla 1. Herramienta por resultado esperado

1.6.2. Herramientas

- SPARQL

Es un lenguaje de consulta para datos almacenados como RDF. Las consultas contienen patrones de tripletas llamados patrones de grafos básicos, los cuales son tripletas RDF, pero con el sujeto, predicado u objeto como variable. Los patrones de grafo básicos son comparados con los datos RDF, en donde se encuentra un subgrafo que sea equivalente a partir del cual se puedan sustituir las variables por valores. Los resultados de estas consultas pueden ser datos o grafos RDF (“SPARQL Query Language for RDF,” n.d.).

- OWL

La W3C *Web Ontology Language* (*OWL*) es un lenguaje usado para representar conocimiento complejo, grupos de recursos y sus relaciones. Este lenguaje expresa el conocimiento de tal manera que puede ser entendido por las computadoras.

OWL es usado para lo siguiente:

- Formalizar un dominio definiendo clases y propiedades de esas clases.
- Definir elementos y establecer sus propiedades.
- Razonar sobre esas clases e individuos en medida que la semántica de OWL lo permitan.

Es necesario que se especifique el sentido de los recursos descritos en la Web,

de esa forma, a través de la interpretación, se obtiene el significado de la data.

Así, *OWL* se basa en una ontología para especificar como se derivan consecuencias lógicas (“OWL Web Ontology Language Overview,” n.d.).

- Apache Jena

Apache Jena es un framework para la construcción de aplicaciones de Web Semántica y *Linked Data*. Esta herramienta contiene diversas *APIs* que permiten procesar información contenida en ontologías, de las cuales se utilizarán la de ontologías, *RDF* y *SPARQL*. Esta herramienta permite la integración con el lenguaje de programación Java. Adicionalmente, Jena provee una herramienta denominada ARQ, que permite un manejo automatizado de sentencias de SPARQL para elaborar consultas complejas con mayor facilidad (“Apache Jena,” n.d.).

- Eclipse

Eclipse es un *IDE* del lenguaje de programación Java, permite incluir funcionalidades adicionales mediante la instalación de plugins y es uno de los IDE más utilizados y estables, por lo que cuenta con el soporte y documentación necesaria para realizar este proyecto. Se utilizará la última versión de este IDE para *Java SE (Standard Edition*), *Eclipse Neon* (“Eclipse - The Eclipse Foundation open source community website,” n.d.).

- Spring Framework

Spring es un framework para el desarrollo de aplicaciones de código abierto para Java. Spring es modular, es decir, se pueden utilizar sólo las partes que uno necesita, de tal manera que no se genera una dependencia de la solución completa y permite la integración con distintas herramientas para el desarrollo de aplicaciones web (Johnson et al., 2004).

Para la implementación de la solución, se utilizará el framework Spring en conjunto con el IDE Eclipse, de la forma de un proyecto Maven, incluyendo las

dependencias necesarias en el archivo de configuración del proyecto.

1.6.3. Metodologías

- Revisión sistemática

Una revisión sistemática es una metodología para identificar, evaluar e interpretar todas las investigaciones realizadas relevantes para una pregunta de investigación o área de interés, estos estudios son denominados primarios

Existen varias razones para realizar una revisión sistemática, de las cuales destacan las siguientes:

- Como una forma de resumir la existencia de evidencia sobre un tratamiento o tecnología
- Para identificar espacios no cubierta en investigaciones actuales, de manera que se puedan sugerir áreas para investigaciones futuras.
- Para proveer un framework con el fin de posicionar apropiadamente nuevas actividades de investigación

El proceso de revisión

La revisión sistemática se divide principalmente en tres fases: planear de la revisión, conducir la revisión y reportar la revisión.

Etapas del planeamiento:

- Identificación de la necesidad de la revisión
- Comisionar una revisión
- Especificación de la pregunta de investigación
- Desarrollo de un protocolo de revisión
- Evaluación del protocolo de revisión

Etapas de la conducción de la revisión:

- Identificación de investigación
- Selección de estudios primarios
- Estudio de la calidad
- Extracción y monitoreo de datos
- Síntesis de datos

Etapas del reporte de la revisión:

- Especificación de mecanismos de diseminación
- Dar un formato al reporte principal
- Evaluación del reporte. (Keele, 2007)

- RUP (Rational Unified Process)

RUP es una metodología de desarrollo de software, desarrollado por una división de IBM, Rational Software Corporation, cuyo propósito es ser adaptado según las necesidades del equipo de desarrollo.

RUP es un enfoque de desarrollo de software que es iterativo, centrado en arquitectura e impulsado en casos de uso. Es un proceso de software bien definido y estructurado, establece claramente quien es responsable de qué, cómo se hacen las cosas y cuando se tienen que hacer, también provee una estructura bien definida para el ciclo de vida de un proyecto, articulando hitos principales y puntos de decisión. Adicionalmente, RUP también es un producto de proceso, que provee un marco de trabajo personalizable de procesos para ingeniería de software. El producto RUP soporta personalización y autorización de procesos, así como una variedad de procesos o configuración de procesos pueden ser elaboradas a partir de esta.

El proceso consta de dos dimensiones, repartidas en dos ejes: el eje horizontal representa el tiempo y muestra el aspecto dinámico del proceso expresado en términos de ciclos, fases, iteraciones e hitos. El eje vertical representa el aspecto estático del proceso, como es descrito en término de actividades, artefactos,

trabajadores y flujos de trabajo. El detalle de las dimensiones puede ser apreciado en la imagen 2.

El ciclo de vida de software se encuentra dividido en ciclos, en donde cada ciclo abarca una nueva generación del producto final, adicionalmente, cada ciclo dividido en cuatro fases consecutivas:

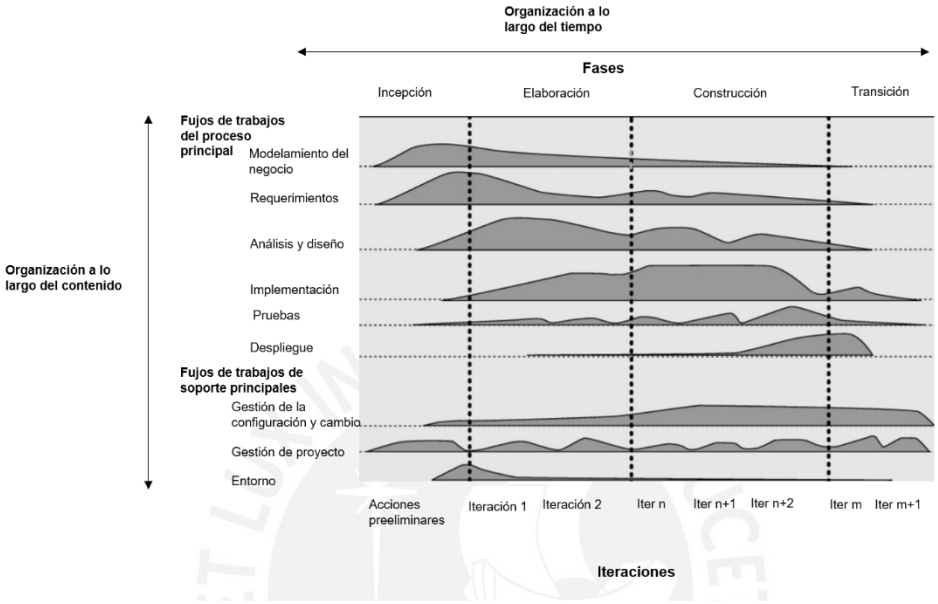

Ilustración 2. Estructura de dos dimensiones de RUP

- Incepción

Primera fase del ciclo de vida de RUP, se enfoca en comprender el alcance del proyecto, para lograrlo se debe identificar todas las entidades que interactuarán con el sistema (actores) y definir la naturaleza de esta interacción en un alto nivel, lo cual involucra identificar todos los casos de uso y describir los más importantes.

- Elaboración

En esta fase lo que se busca es definir la arquitectura del sistema para brindar una base estable al diseño y esfuerzo de implementación en la siguiente fase. Las decisiones con respecto a la arquitectura deben ser tomadas con un entendimiento de todo el sistema: su alcance, requerimientos funcionales y no funcionales.

- Desarrollo

La fase de desarrollo o construcción busca implementar e integrar todos los componentes restantes, así como probar cada funcionalidad. En cierta manera, es un proceso de manufactura en donde el énfasis es colocado en administrar recursos y controlar operaciones para optimizar costos, cronogramas y calidad.

- Transición

El propósito de la fase de transición es entregar el producto de software a la comunidad de usuarios. Una vez realizado esto, se requerirán desarrollar nuevas entregas, corregir algunos problemas o completar las funcionalidades que fueron pospuestas.(Rational Software, 1998)

Para este proyecto se eligió RUP debido a su enfoque centrado en metas, las que se pueden medir según lo planificado en el cronograma, a diferencia de otras metodologías ágiles, que no establecen plazos definidos a cumplir como se exige en un proyecto de tesis.

1.7. Alcance

Este proyecto contempla la implementación de un navegador de Linked Data, que corresponde al área de ciencias de la computación, específicamente Web Semántica y Linked Data. La información utilizada será la publicada en los repositorios científicos, que contengan datasets en el dominio de ciencias de la computación; dado que se necesita el uso de ontologías específicas, la información se limitará únicamente a este dominio. El software implementado será accesible por los usuarios vía web desde un navegador tradicional.

1.8. Riesgos

A continuación, se presentan los principales riesgos identificados durante el desarrollo del proyecto en la siguiente tabla, el impacto que se les asignó y las medidas que permitirán mitigarlos.

Riesgo identificado	Impacto en el proyecto	Medida correctiva para mitigar
Alta complejidad de las herramientas utilizadas en el proyecto genere retrasos en el cumplimiento del cronograma o una baja calidad del entregable.	Medio	Aprendizaje de la herramienta con anticipación mediante la documentación disponible.
Retrasos en la entrega de los avances al asesor genere un incumplimiento del cronograma o una baja calidad del entregable.	Medio	Reuniones virtuales con el asesor, comunicación mediante otros medios.
Problemas de salud que generen retrasos o impidan la implementación del proyecto	Medio	Realizar la planificación tomando en cuenta imprevistos de este tipo.
Cambio o actualización de las herramientas utilizadas en el proyecto genere retrasos o problemas por falta de soporte.	Medio	Revisión periódica de las versiones de las herramientas.

Tabla 2. Riesgos identificados en el proyecto de tesis

1.9. Justificativa y viabilidad del proyecto

1.9.1. Justificativa

La web es un gran espacio de información, de donde esta se obtiene a través de motores de búsqueda, sin embargo, dado que la búsqueda se limita a encontrar el mejor resultado posible de un gran número de documentos, estos pueden contener o no la información que el usuario desea. Por otra parte, si no se es experto en la materia en cuestión, los motores de búsqueda no resultan de gran utilidad al no poder optimizar las cadenas de búsqueda. Los resultados probablemente no sean los óptimos, además de consumir tiempo valioso para el usuario.

La Web Semántica intenta resolver este problema, dado que la información se encuentra estructurada por dominios de conocimiento de tal manera que pueda ser procesada tanto por personas como por máquinas, como en un principio la web había sido diseñada. Pero para poder aprovechar toda esta información se requiere conocimiento técnico, en el lenguaje de consulta SPARQL y en RDF y XML para poder interpretar las estructuras obtenidas de los datasets consultados.

Este proyecto busca contribuir a solucionar esta problemática implementando un navegador de Linked Data en el dominio de ciencias de la computación, para que los especialistas del área tengan acceso a toda la información contenida en datasets escogidos relacionados a este campo, de tal manera que puedan obtener información relevante mediante una herramienta que no requiere conocimientos técnicos en las tecnologías de la Web Semántica.

1.9.2. Viabilidad

- **Viabilidad técnica**

Las herramientas necesarias para la implementación de este proyecto son de código abierto, tanto el entorno de desarrollo Eclipse y el framework de desarrollo web Spring, que en conjunto serán utilizados para implementar la interfaz gráfica, como lo proporcionado por el framework Apache Jena, que brinda un API para RDF y SPARQL, necesarios para acceder a las estructuras RDF obtenidas de los repositorios seleccionados para el proyecto.

De igual manera, todos los repositorios consultados brindan SPARQL endpoints de manera libre, por lo que la solución se podrá implementar de manera web. Adicionalmente, se pueden obtener los archivos RDF de dichos repositorios, lo que permite que la solución también pueda funcionar sin conexión a internet.

En cuanto a conocimientos técnicos, pese a no haber llevado cursos relacionados a Linked Data y Web Semántica ni contar con experiencia previa, se cuenta con la orientación del asesor de esta tesis, quien es especialista en

estos temas; por otra parte, todas las documentaciones de las herramientas necesarias se pueden encontrar en los sitios webs, por lo tanto, se concluye que este proyecto es técnicamente viable

- **Viabilidad temporal**

Para la implementación del proyecto de tesis se estimaron una serie de tiempos clasificados por resultados esperados, los cuales se presentan en la siguiente tabla.

Resultado esperado	Duración (semanas)
RE1	2
RE2	2
RE3	1
RE4	2
RE5	2
Total	9

Tabla 3. Tiempo estimado de implementación de los resultados esperados

En base a que el tiempo estimado concuerda con el cronograma de avances planificado para el curso de Tesis 2, se concluye que el proyecto es temporalmente viable.

- **Viabilidad económica**

Todas las herramientas determinadas para la realización de este proyecto son gratuitas. Todas las fuentes bibliográficas investigadas son de libre acceso, además se cuenta con una computadora personal más las brindadas por los laboratorios de la Universidad, así también como con acceso a internet, por lo

que no se tendrá que incurrir en algún gasto adicional. Por lo tanto, se concluye que este proyecto es económicamente viable.

1.10. Análisis de necesidades

El presente proyecto sigue la línea de investigación referente a los estándares de la Web semántica; sin embargo, también se hace uso de modelos de conocimiento como lo son las ontologías, por lo que se tiene interacción con el área de ingeniería del conocimiento. En ese caso, se contará con el apoyo de un doctor en ingeniería del conocimiento, quien con su experiencia en dicha área orientará el progreso del proyecto.

Capítulo 2. MARCO CONCEPTUAL Y ESTADO DEL ARTE

2.1. Marco conceptual

En este apartado se definirán los términos necesarios para una mejor comprensión del problema expuesto. Se empezará presentando el concepto de Web semántica, seguido de las tecnologías usadas como soporte. Luego, se explicará lo que es *Linked data* y cómo es que a través de ella se estructuran datos, para posteriormente publicarlos como abiertos (*Linked open data*). Finalmente, se define ontología y la aplicación de la inferencia sobre ella para obtener nuevo conocimiento.

2.1.1. Web semántica

La Web semántica es denominada como Web de data ("Semantic Web roadmap," n.d.). Inicialmente, la Web fue ideada como el espacio de información en el cual se facilitará la comunicación entre humanos, pero además, un lugar en donde las máquinas pudieran intervenir; sin embargo, el diseño y estructura de la Web no facilitan lograr lo último ("W3C Semantic Web Activity," n.d.). Es allí en donde surge el enfoque de expresar la información en un formato estándar que sea procesable por máquinas. Para ello, se emplean diversos conceptos, herramientas y tecnologías que permiten organizar toda esa data y procesarla, obteniendo como resultado una mayor cantidad de información y conocimiento.

La Web Semántica se encuentra regida por 6 principios:

- Principio 1: Todo puede ser identificado por URIs

Las personas, lugares y cosas en el mundo real pueden ser definidos en la Web Semántica mediante una variedad de identificadores. Cualquier usuario puede crear un URI y decir que identifica algo en el mundo real.

- Principio 2: Los recursos y enlaces pueden tener tipos

La Web actual consiste en recursos y enlaces. Los recursos son documentos orientados al consumo humano. En la Web Semántica, estos recursos contienen tipos que brindan información que puede ser procesada por una máquina.

- Principio 3: La información parcial es tolerada

La Web actual no tiene límites, sacrifica integridad de los enlaces por escalabilidad. De la misma manera, la Web Semántica tampoco tiene límites, cualquiera puede decir cualquier cosa de un recurso y crear diferentes tipos de relaciones entre estos. Si un recurso deja de funcionar, las herramientas de Web Semántica deben tolerar el fallo de dicho recurso y seguir funcionando.

- Principio 4: No existe una necesidad de verdad absoluta

No todo lo que se puede encontrar en la Web es verdadero y la Web Semántica no cambia esto. La confiabilidad de la información depende de las fuentes de obtención de esta independientemente de las tecnologías aplicadas con la Web Semántica.

- Principio 5: La evolución es soportada

La Web Semántica permite combinar información disponible en la Web para expandirla de la misma manera en que se expande el entendimiento humano. Esto permite que se agregue nueva información sin que la actual se tenga que modificar.

- Principio 6: Diseño minimalista

El objetivo de la Web Semántica es no estandarizar más de lo necesario. Esto permite la implementación de aplicaciones simples basadas en tecnologías estandarizadas, así como también investigación para futuros planes complejos.

Además, la Web semántica presenta una arquitectura en la que se representa la jerarquía de las tecnologías estándares empleadas. Se puede observar, en la Imagen 3, que la capa más baja contiene la tecnología URI, usada para identificar los recursos, y Unicode, que permite trabajar con texto en diversos lenguajes. Esta representación única es necesaria para brindar soporte a las capas superiores.

En la capa dos, XML es utilizada para estructurar documentos de manera legible. La capa tres presenta RDF, que permite representar la información en forma de grafo. Sobre ella se encuentra la capa de Ontología, la cual añade más restricciones a las establecidas en RDF. La capa lógica brinda la oportunidad de deducir relaciones que no han sido descritas explícitamente. Finalmente, las capas de pruebas y confianza garantizan que la información obtenida provenga de fuentes confiables y de la inferencia a través de la lógica formal.

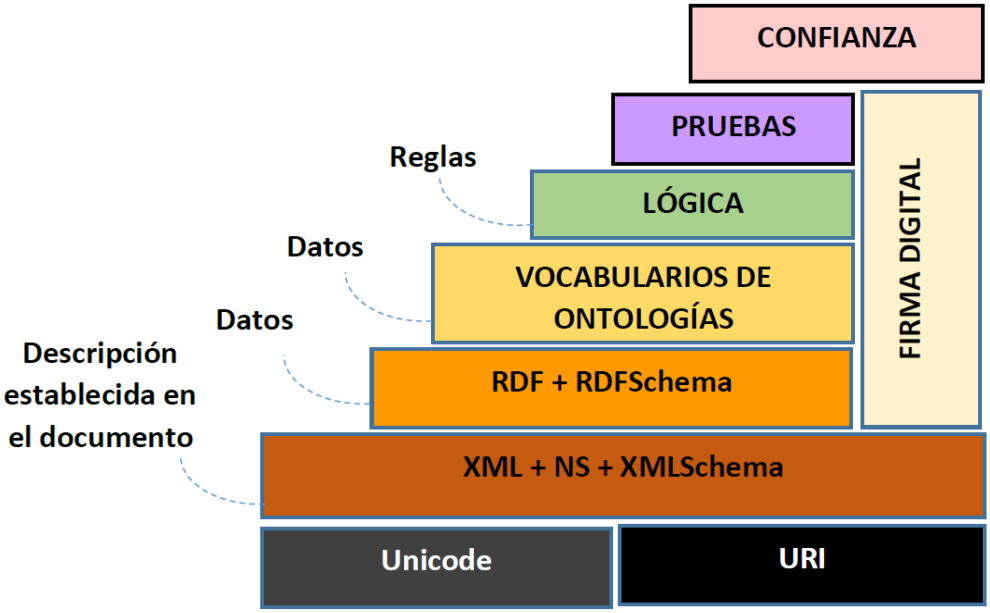

Ilustración 3. Capas de la Web Semántica

Por otro lado, para lograr el enriquecimiento de la información, la Web semántica hace uso de herramientas como URIs, RDF, SPARQL y OWL, las cuales serán descritas a continuación:

2.1.2. URI

Un URI es una cadena de caracteres que representa a un recurso. En la Web semántica, es usada para representar cosas en el mundo físico como personas y lugares. De esa forma, se evita ambigüedades a través de la abstracción del lenguaje natural, brindando una forma estándar para hacer referencia a algún recurso (“RDF - Semantic Web Standards,” n.d.). En la imagen 4 se ilustra un ejemplo, en este caso el URI http://www.ejemplo.com/idEnfermedad/1 representa el recurso de una enfermedad que posee el identificador 1.

Un URI tiene las siguientes propiedades:

- Uniforme

Al ser uniformes pueden ser reutilizados en distintos contextos, son transparentes a los mecanismos que utilizan para ser accedidos, lo que permite escalabilidad, nuevas aplicaciones o protocolos y una cantidad ilimitada de URIs sin mayores problemas

- Recurso

Un recurso puede ser cualquier entidad u objeto, su alcance no se encuentra limitado, podría ser una persona, una ciudad, un servicio, e incluso una colección de recursos. Básicamente, un recurso es algo que puede ser identificado por una URI dentro de este contexto.

- Identificador

La definición de identificador hace referencia lo que puede distinguir algo identificado respecto a todo lo demás, hablando de recursos, independientemente de la forma en la que lo realice.

2.1.3. RDF

Es un modelo estándar para el intercambio de datos en la Web. RDF sigue el principio de usar URIs para representar relaciones y los recursos relacionados, a estos enlaces se le denominan triple ("Query - W3C," n.d.). Esta estructura forma un grafo, como el mostrado en la Imagen 4, en donde los recursos son representados por los nodos y las relaciones por las aristas ("Linked Data - Design Issues," n.d.).

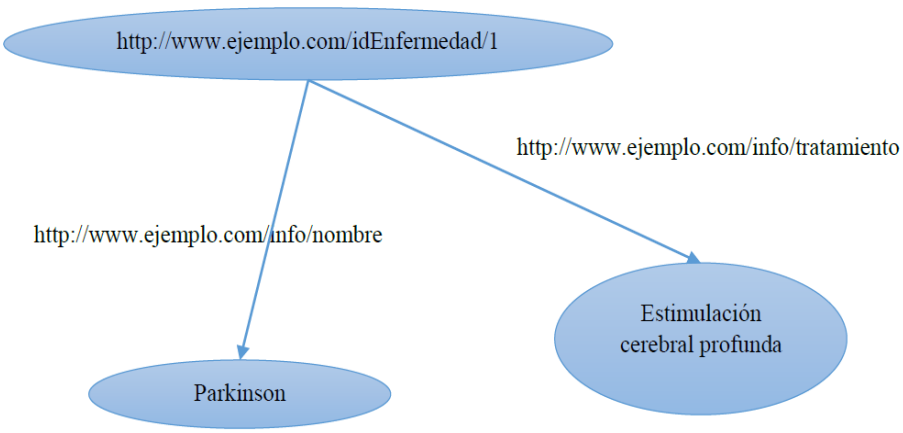

Ilustración 4. Ejemplo de grafo RDF

2.1.4. Linked Data

En la Web semántica, no solo es importante que una gran cantidad de data esté disponible en un formato estándar, sino que, además, deben estar presentes relaciones entre ellas. *Linked Data* se define como la colección de conjunto de datos relacionada.

Tim Berners-Lee postula los siguientes 4 principios:

1) Usar URIs para nombrar cosas y representar recursos.
2) Usar HTTP URIs para que puedan ser interpretados y desreferenciados.
3) Cuando se hace referencia a un URI, se debe brindar información útil, mediante el uso de estándares (RDF, SPARQL).
4) Se debe hacer referencia a otros URIs, ya que es necesario que la data esté conectada (Bizer et al., 2009).

2.1.5. Linked Open Data

Es definida como *Linked Data* cuyo contenido es publicado bajo licencias abiertas. De esta manera, se agrega un quinto principio a los cuatro principios de *Linked Data*, contenido abierto.
Un caso de aplicación es el de DBpedia, que tiene como objetivo convertir el contenido de Wikipedia en conocimiento estructurado (Auer et al., 2007), permitiendo así que se le apliquen técnicas de la Web semántica.

2.1.6. Ontología

El término proviene del campo de la filosofía, en donde trata con la naturaleza de lo que existe, así como su agrupamiento y relaciones. Una ontología es una especificación explícita de una conceptualización (Genesereth & Nilsson, 1987), en donde conceptualización es definida como los objetos, conceptos y otras entidades en una determinada área de interés, además de las relaciones entre ellos (Peirce, Hartshorne, & Weiss, 1932).

Actualmente, las ontologías son ampliamente usadas en áreas como ciencias de

la computación, medicina, entre otras; en las que son utilizadas para representar conocimiento y descubrir nuevas relaciones a través de la inferencia.

2.1.7. Navegador de Linked Data

Tipo de aplicación de Linked Data que, a diferencia de los navegadores tradicionales, que permiten la navegación entre documentos HTML, los navegadores de Linked Data permiten al usuario navegar entre fuentes de información mediante enlaces, expresados en formato RDF. El principio de estos navegadores se basa en que los datos brindan a una interfaz humana más oportunidades y retos de los que podría brindar un navegador de hipertexto tradicional (Bizer et al., 2009).

2.1.8. Motor de búsqueda de Linked Data

Al igual que los motores de búsqueda de *Linked Data*, son un tipo de motores de búsqueda que obtienen rastrean información enlazada de la Web como enlaces de RDF y brindan funcionalidades de consulta sobre este tipo de información. Se pueden clasificar en dos tipos:

Motores de búsqueda orientados a las personas. Proveen un motor de búsqueda basado en palabras claves relacionadas a un tema de interés del usuario, que devuelve una lista de resultados. Adicionalmente, permiten al usuario explorar y explotar la información obtenida a partir de su estructura.

Índices orientados a aplicaciones. Consisten en APIs que permiten a las aplicaciones descubrir documentos RDFs en la Web que referencian un cierto URI o contienen alguna palabra clave con el objetivo de que todas las aplicaciones de *Linked Data* no tengan que construir su propia infraestructura y puedan reutilizar estos motores de búsqueda (Bizer et al., 2009).

2.2. Estado del arte

2.2.1. Introducción

El objetivo del estado del arte es dar a conocer diferentes propuestas encontradas que presenten una plataforma que permita la obtención de información relevante aplicando los conceptos de la Web Semántica. Las plataformas que presentarán corresponden a Buscadores Semánticos y Navegadores Semánticos o de *Linked Data* debido a que son las que más se asemejan a la solución planteada en la problemática.

2.2.2. Método usado en la revisión del estado del arte

El método aplicado en la revisión del estado del arte fue el de revisión sistemática de la actual bibliografía con la que se dispone. Una revisión sistemática es una metodología para identificar, evaluar e interpretar todas las investigaciones disponibles que son relevantes para una pregunta de investigación (Keele, 2007). El motivo por el que se usa la revisión sistemática es que con ella se evita la parcialidad que se tiene con las revisiones tradicionales.

Para la realización de la revisión se tomó en cuenta lo siguiente:

- Como preguntas de investigación, se establecieron las siguientes:
    - ¿De qué manera un navegador de *Linked data* mejora la experiencia de obtención de información relevante por parte del usuario?
    - ¿En qué difieren los resultados de búsqueda entre un navegador de *Linked data* y un navegador de documentos de hipertexto tradicional?
    - ¿Existen aplicaciones que utilicen tecnologías de la Web Semántica y Linked Data para obtener información relevante para el usuario? ¿Se necesitan conocimientos técnicos para poder utilizarlas?
- Para la lista con los términos a buscar, se establecería que la búsqueda se realice por títulos de documentos, en la Tabla 1 se muestran las cadenas que

se utilizaron para dicha búsqueda.

Cadenas de búsqueda	
1	“Linked data” Y (“browser” O “search engine”)
2	“Semantic web” Y (“browser” O “search engine”)

Tabla 4. Cadenas de búsqueda

- Las fuentes seleccionadas para llevar a cabo las búsquedas fueron Scopus, *Science Direct* y *IEEE Xplore.*

- Para seleccionar los estudios se siguió un proceso iterativo e incremental.

- Los criterios de inclusión fueron el análisis del resumen, el título y las palabras clave indexadas.

- El criterio de exclusión restringía la búsqueda a publicaciones comprendidas entre 2006 y la actualidad. Algunas de los estudios hallados no son tan recientes; sin embargo, fueron considerados por el reconocimiento con el que contaban sus autores.

2.3. Estudio N°1: Navegador de Web semántica orientado a usuarios sin experiencia en RDF

Este proyecto propone un buscador de Web Semántica que sea de fácil uso, orientado para usuarios novatos o sin experiencia previa en el uso de RDF u OWL, tecnologías utilizadas por la Web Semántica.

Para ello, se plantea tomar como punto de partida un URI de un documento RDF y mostrar la información obtenida en una estructura de grafo, usando círculos y flechas como se muestra en la Imagen 5. De esta forma, a diferencia de los navegadores tradicionales, que permite hacer el recorrido entre documentos, en este proyecto se plantea la navegación a través de datos.

Para la navegación, el usuario simplemente tiene que seleccionar cada nodo y flecha con un *click*, como resultado el grafo mostrará la información correspondiente a ese URI/URL; es decir, se obtienen los URIs relacionados al seleccionado y se muestran estos enlaces en la nueva estructura de grafo actualizada(Kim, Yoo, & Park, 2012).

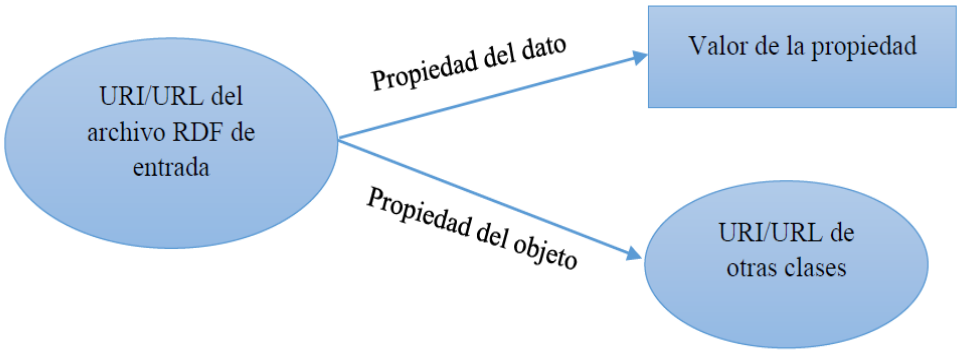

Ilustración 5. Formato abstracto de grafo

La solución planteada en este proyecto muestra resultados que son fáciles de visualizar e intuitivos para personas sin conocimientos ni experiencia en RDF, sin embargo, para un usuario que no posee dichos conocimientos, no será sencillo obtener el URI del recurso a partir del cual desea obtener información relevante.

2.4. Estudio N°2: Motor de búsqueda de Linked data en un dominio específico

Con este estudio se introducen los requerimientos necesarios para el desarrollo de motores de búsqueda orientados a los datos científicos. El proceso de implementación incluye cuatro principales procesos: publicación de datos/metadatos científicos, recuperar datos/metadatos científicos, búsqueda de datos científicos y descubrimiento de enlaces entre ellos.

Los usuarios de este proyecto pueden ser tanto usuarios con o sin experiencia en tecnologías de la Web Semántica, específicamente SPARQL, ya que se le permite al usuario buscar información tomando como entrada una consulta en este lenguaje.

Por otra parte, esta herramienta, al trabajar con datos enlazados, provee información estructurada, cuyas propiedades se encuentran relacionadas al dominio de la búsqueda. Esto es de gran importancia al consultar información científica, en donde los motores de búsqueda tradicionales no siempre brindan los resultados más precisos (Shen, Hou, Li, & Li, 2012).

2.5. Estudio N°3: Buscador basado en agentes inteligentes y OWL

Este proyecto propone la creación de una comunidad de agentes inteligentes, los cuales llevan a cabo el papel de rastreadores en búsqueda de vecindades de Web semántica, una base de datos semántica para almacenar todos los datos de diversas ontologías, un mecanismo de consulta que permita hacerlas en OWL y un algoritmo para el *ranking* de los resultados; es decir, para el establecimiento del orden basado en las relaciones semánticas.

A diferencia de los motores de búsqueda actuales, WebOWL no se enfoca ni en páginas ni en ontologías completas, sino que se centra en las entidades que se encuentran en dichas ontologías. Para el *ranking*, emplea un algoritmo que asigna distinto poder de clasificación.

Con respecto al uso, WebOWL necesita de un ejemplo para poder realizar la consulta; es decir, el usuario provee la clase de la que necesita información. En el ejemplo del proyecto, se buscaba información sobre la clase herbívoros, por lo que en la ontología “personas+mascotas” estaría definida como animal que come hojas, por lo que se añadió esta propiedad para la consulta.

Este estudio muestra cómo trabajan las ontologías y le dan un contenido semántico a la información, sin embargo, se necesita un conocimiento previo del lenguaje OWL para poder realizar búsquedas así como el conocimiento de las ontologías asociadas a las propiedades que el usuario ingresa (Batzios & Mitkas, 2012).

2.6. Estudio N°4: Navegador Semántico como extensión de un navegador tradicional

Tabulator es un Navegador de Web Semántica que permite la navegación entre contenido RDF. Consiste en la extensión de un navegador tradicional añadiendo las funcionalidades requeridas para trabajar como un navegador genérico de *Linked Data*.

Funciona en 2 modalidades: exploración y análisis. En el modo de exploración, el usuario ingresa un URI y el navegador le muestra información en un grafo RDF, permitiendo la navegación entre los nodos, pudiendo expandir un nodo para obtener más información sobre este. En modo de análisis, el usuario selecciona ciertos campos para definir un patrón y consulta a Tabulator para obtener todos los ejemplos de dicho patrón. Los resultados de la consulta se muestran en forma de una tabla, calendario o mapa a partir de datos en RDF.

El trabajo logra mejorar la experiencia de obtención relevante por parte del usuario, presenta un navegador que utiliza tecnologías de la Web Semántica para lograr su propósito y ha sido implementado como una extensión de un navegador tradicional, por lo que su instalación no debería presentar dificultades (Berners-Lee et al., 2006).

2.7. Conclusiones

Los estudios presentados en el estado del arte demuestran que existen aplicaciones que aprovechan las tecnologías de la Web Semántica para obtener información relevante para el usuario, sin embargo, una gran parte de ellos requiere experiencia o conocimiento en dichas tecnologías, como navegación entre grafos RDF, elaboración de consultas en lenguaje SPARQL o conocimiento de ontologías ampliamente utilizadas en el lenguaje OWL. Por otra parte, hay proyectos que sí se han elaborado para usuarios sin estos conocimientos, con la finalidad de mostrar la información de una manera gráfica y simple.

La búsqueda de información al consultar la Web Semántica la presenta estructurada, de manera que pueda ser entendida y procesada por las máquinas,

según el dominio de cada recurso gracias al uso de ontologías, a diferencia de consultar en la Web tradicional, que se limita a la búsqueda de información en documentos. Esta ventaja se puede apreciar especialmente al buscar información científica, en donde una búsqueda sintáctica no siempre devuelve los resultados esperados.

Capítulo 3. OBJETIVO ESPECÍFICO 1: DISEÑAR E IMPLEMENTAR UN MECANISMO DE PROCESAMIENTO DE CONSULTAS DEL DOMINIO.

3.1. Introducción

Los motores de búsqueda tradicionales requieren de una consulta en forma de cadena de búsqueda ingresada por el usuario que será procesada y a partir de la cual se obtendrán resultados que se esperan que sean relacionados a dicha consulta. La eficiencia de la búsqueda dependerá de que tan bien elaborada se encuentre la cadena de búsqueda, sin embargo, dado que el usuario no siempre es un especialista en el área a investigar, realiza búsquedas con consultas que no brinden los resultados óptimos, perdiendo tiempo navegando entre una gran cantidad de documentos hasta obtener la información que necesita.

Mediante la extracción de la información de la nube de *Linked Data* se busca mejorar dicha experiencia para el usuario y dada las funcionalidades de los motores de búsqueda actuales y la experiencia de los usuarios en dichas herramientas, en este objetivo específico se propone trabajar inicialmente de la misma manera, solicitándole al usuario una cadena de búsqueda inicial, que será procesada para luego obtener el dominio de la consulta en base a dicha cadena.

3.2. Resultado Esperado 1: Módulo de procesamiento de lenguaje natural para procesar la consulta

Para esta etapa del proyecto se ha trabajado con ontologías previamente definidas relacionadas al dominio, las cuales limitarán el alcance de los temas de búsqueda accesibles para el usuario, dado que los conceptos estarán limitados por las propiedades definidas en las ontologías.

Este resultado busca procesar la cadena de búsqueda ingresada para así obtener la ontología (entre las previamente definidas) cuyos conceptos definidos en sus propiedades se acerquen más a la información requerida por el usuario. Las consultas pueden estar en la forma de una simple palabra, así como una oración, que el usuario espera que la herramienta responda, brindándole información relevante. Para poder procesar este tipo de consultas se utilizaron

técnicas de procesamiento de lenguaje natural, dado que la cadena puede estar en la forma de una oración completa, a partir de la cual se deben extraer los conceptos que se desean buscar en la Web semántica.

3.2.1. Resultado alcanzado

El mecanismo comienza con el procesamiento de la cadena de búsqueda, dicho proceso consistió en transformar cada palabra a su forma base y clasificarlas para luego escoger el tipo de palabras que se desean utilizar. Para esto se utilizó la librería de Procesamiento de Lenguaje Natural (*NLP*) de la Universidad Stanford, Stanford Core NLP que contiene los pasos básicos de NLP, desde tokenización hasta correferencia (o co-indexación) (Manning et al., 2014); utilizando las funciones provistas por esta librería se pudo transformar cada palabra de la consulta a su forma lema o canónica. Una vez realizado esto, se procedió a clasificar las palabras según su la estructura gramatical de la oración, para esto, se utilizó un *Part of Speech (POS) Tagger*, también provisto por la librería NLP de Stanford. Esta herramienta permite asignar una etiqueta a cada palabra de una oración provista, la cual corresponde a su estructura gramatical (Toutanova & Manning, 2000). Luego de dicha transformación se obtuvieron todos los sustantivos de la oración, dado que son los elementos más probables que estén asociados a un concepto.

Por otra parte, también se tuvo que tener en consideración el caso en que el usuario ingresa palabras mal digitadas, si se realiza una búsqueda por comparación de cadenas con palabras inexistentes contra las palabras obtenidas anteriormente no se encontraría ningún resultado. Para mitigar este posible riesgo se realizarán búsquedas por similitud de palabras para tomar en cuenta. Esto cobra importancia si se toma en cuenta que la velocidad de tipeo promedio de una persona es de 41 palabras por minuto, lo cual lleva a un promedio de un 92% de precisión, es decir, de cada 100 palabras tipeadas, 8 contienen errores (“Average typing speed,” n.d.).

Para implementar este tipo de búsqueda se utilizó el algoritmo de Distancia de Levenshtein para medir la distancia entre palabras. El algoritmo encuentra el set de inserciones, eliminaciones o sustituciones menos costoso necesario para transformar una cadena en la otra (Heeringa, 2004).

3.2.2. Pruebas

- Etiquetado de palabras

  Se realizó una consulta relacionada al dominio de Ciencias de la Computación para probar este mecanismo, la consulta fue: *"How does machine learning techniques work?"*. Tras utilizar las funciones provistas por el POS Tagger de la librería NLP se obtuvieron las siguientes etiquetas:

  How: WRB (Wh-adverb: Adverbio que en inglés comienza con las letras wh)

  Does: VBZ (Verbo en tercera persona, singular, presente)

  Machine: NN (Sustantivo en singular)

  Learning: NN (Sustantivo en singular)

  Techniques: NNS (Sustantivo en plural)

  Work: VB (Verbo en forma base)

  Una vez realizado el etiquetado se puede analizar la consulta en base al contenido sintáctico de esta para poder obtener las posibles propiedades. De esta manera, se escogieron todas las variedades de sustantivos para procesarlas en la siguiente fase; en este caso, las etiquetas seleccionadas fueron NN (Sustantivo en singular) y NS (Sustantivo en plural).

- Obtención de la forma canónica

  Para que la búsqueda de propiedades sea más efectiva, se procedió a reducir las palabras a su forma canónica o lema. Esto permite que en los siguientes pasos se realicen búsquedas comparando cadenas y la probabilidad de encontrar las propiedades asociadas a cada palabra sean mayores. Como solo se trabajará con sustantivos, se eligieron las palabras: "Machine", "Learning" y "Techniques", el resultado obtenido fue el siguiente:

  Machine: Machine (no se aplicó ninguna transformación).

Learning: Learn (se transformó de un sustantivo en singular a un verbo en infinitivo).

Techniques: Technique (se transformó de plural a singular).

- Búsqueda por similitud

Se implementó la comparación por distancia basada en la Distancia de Leveshtein de tal manera que al comparar dos palabras se obtiene un ratio de similitud entre 0 a 100, en donde la comparación de dos palabras idénticas equivale a un ratio de 100.

Se procedió a comparar las palabras obtenidas en la búsqueda anterior con las mismas palabras, pero cambiando entre 1 a 3 letras. Los resultados fueron los siguientes:

Palabra Inicial	**Palabra Transformada**	**Letras de diferencia**	**Ratio de similitud**
Machine	Machino	1	86
Machine	Macxino	2	71
Machine	Nacxino	3	57
Learning	Learming	1	88
Learning	Rearming	2	75
Learning	Rearninh	3	63
Techniques	Pechniques	1	90
Techniques	Peqhniques	2	80
Techniques	Peqhniquez	3	70

Tabla 5. Resultados de comparación entre palabras similares

Según los resultados mostrados en la tabla 5, se determinó que una similitud de 70 o más sería adecuado para determinar que dos palabras son parecidas y su ocurrencia puede indicar un posible error de tipeo por parte del usuario.

Esta funcionalidad se integrará al proyecto en caso de que la búsqueda regular no haya devuelto ningún resultado, considerando la posibilidad de que el usuario haya cometido algún error al digitar la cadena de búsqueda. En caso de que esto

suceda, se procederá a comparar los términos con resultados parecidos según la similitud entre palabras priorizando aquellos valores con mayor valor de cercanía de distancia.

Capítulo 4. OBJETIVO ESPECÍFICO 2: DISEÑAR E IMPLEMENTAR UN MECANISMO QUE PERMITA OBTENER Y PROCESAR INFORMACIÓN DE LA NUBE DE LINKED OPEN DATA.

4.1. Introducción

Una vez obtenida la ontología en el objetivo anterior, el siguiente paso será obtener el recurso en la Web Semántica que contenga la información indicada en las propiedades de dicha ontología.

4.2. Resultado Esperado 2: Estructura RDF asociada al recurso indicado por la URI obtenida.

La estructura *RDF* será creada y poblada en base a las propiedades de la ontología y la información contenida en el *dataset* consultado; para realizar dicha consulta, se utilizará el lenguaje de consulta *SPARQL*. El *framework* Apache Jena brinda un API para trabajar con estructuras *RDF* y realizar consultas en lenguaje *SPARQL* utilizando el lenguaje de programación Java. Adicionalmente, con el API de Jena, se utilizó ARQ, otra herramienta para automatizar la creación de consultas SPARQL, que permitió trabajar de manera más estructurada.

4.2.1. Resultado alcanzado

Se utilizó como fuente de información datasets asociados al dominio de Ciencias de la Computación, que contienen principalmente información sobre trabajos académicos y todo lo relacionado a estos. Dichos repositorios son los siguientes:

- (Digital Bibliography and Library Project) DBLP Computer Science Bibliography

DBLP es un repositorio que contiene información bibliográfica sobre trabajos académicos de Ciencias de la Computación, brindado por la Universidad de Trier, en conjunto con el Schloss Dagstuhl - Centro de Informática de Leibniz.

Adicionalmente, el proyecto DBLP recibe apoyo del Instituto de Investigación Alemán, la Asociación Leibniz, la Fundación Klaus Tschira, *Microsoft Research* y la organización VLBDE (*Very Large Data Base Endowment*), como también donantes individuales.

DBLP evolucionó de una pequeña plataforma web experimental a uno de los servicios más populares en la comunidad de ciencias de la computación. Para mayo del 2016 DBLP había indexado más de 3.3 milones de publicaciones, publicadas por más 1.7 millones de autores. Hasta la fecha, ha indexado alrededor de 32000 *journals*, más de 31000 conferencias y más de 21000 monografías. Cuenta con 57674239 tripletas en formato RDF. La bibliografía de DBLP abarca solamente publicaciones de ciencias de la computación, enfocándose en publicaciones internacionales, es decir, en inglés, con algunas excepciones, tomando como criterio de inclusión el mérito científico de la publicación. Algunos de los tópicos que abarca DBLP son los siguientes: Algoritmos, Inteligencia Artificial, Automatización, Bioinformática, Sistemas de Información Compiladores, Arquitectura de Computadores, Gráficos en Computación, Visión Computacional, Minería de Datos, Interacción Persona-Computador, Aprendizaje Máquina, Web Semántica, entre otros.

El portal web de DBLP almacena cuatro tipos diferentes de páginas web: páginas de índice, páginas de flujo de publicación, tabla de páginas de contenido y páginas de persona (“DBLP - Computer Science Bibliography,” n.d.).

- Índices

En el nivel más alto de la jerarquía de la página de DBLP, todos los *journals* se encuentran listados en el índice de *journals*, mientras que las conferencias están listadas en el índice de conferencias, como se puede apreciar en la Imagen 6, ambos índices proveen enlace a páginas de los flujos de publicación

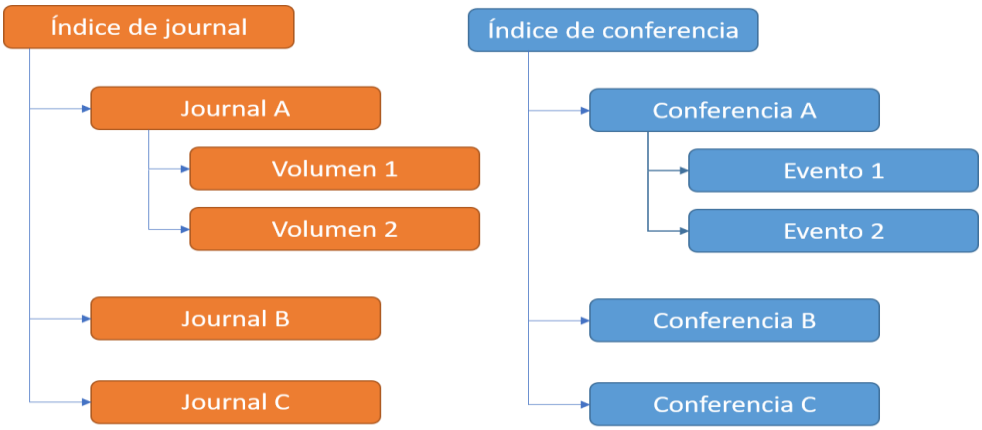

Ilustración 6. Estructura de Índices de DBLP

- Flujos de Publicación

El siguiente nivel de páginas de DBLP son los flujos de publicación o también denominados "lugares". Un flujo de publicación sintetiza los volúmenes de procedimientos de una serie de conferencias como VLDB o los volúmenes de un *journal* como TODS (*ACM Transactions on Database Systems*). Las páginas de flujos de publicación contienen enlaces a las tablas de contenidos de las conferencias o volúmenes de *journals*.

- Tabla de contenidos

La tabla de contenidos de un *journal* o volumen de procedimientos son el nivel más bajo de la estructura web de DBLP, en donde la *metadata* bibliográfica para cada artículo del volumen es listado, es decir, la lista completa de autores, el título, el número de páginas y cualquier enlace disponible.

- Autores

Cuando el nombre de una persona aparece en el sitio web de DBLP, ese nombre provee un enlace a la página del autor, en donde todas las publicaciones identificadas de dicha persona son listadas.

- Institute for Electrical and Electronics Engineers (IEEE) Papers

Repositorio que contiene las publicaciones de la IEEE, Asociación Mundial de Ingenieros dedicados a promover el desarrollo en diversas áreas técnicas. Cuenta con más de 423000 miembros en alrededor de 160 países. Fundada en 1963, IEEE es la más sociedad más grande de profesionales técnicos, diseñada

para servir a dichos profesionales en todos los aspectos de los campos de electricidad, electrónica y computación, como también áreas relacionadas de ciencia y tecnología que comprenden nuestra sociedad moderna. Asimismo, IEEE cuenta con profesionales de las áreas de ingeniería, ciencias de la computación, desarrollo de software, profesionales en tecnología, médicos y muchos otros, sin mencionar al núcleo de ingenieros eléctricos y electrónicos, por lo que la organización ya no se identifica por su nombre completo, excepto en casos legales, y es normalmente referida simplemente como IEEE.

La IEEE se encuentra organizada en: Secciones locales, dentro de sus regiones geográficas; Capítulos, sub-unidades técnicas de una o más secciones de IEEE, compuestos por miembros locales con intereses técnicos similares, pueden incluir ponentes invitados, *workshops* y seminarios y proveen a los miembros de una sociedad oportunidades de *networking* en un nivel local; Sociedades y Consejos Técnicos, que componen divisiones técnicas, IEEE tiene 39 sociedades técnicas que permiten enfocarse a sus miembros en sus profesiones tecnológicas, trabajar con colegas de manera local y en el extranjero, así como también colaborar en proyectos de investigación; ramas estudiantiles en las universidades alrededor del mundo, que permiten a los miembros estudiantes iniciarse en *networking* en sus áreas de interés y futura profesión, contando con más de 3000 Ramas Estudiantiles en más de 100 países y Capítulos de Rama Estudiantil.

Por otra parte, IEE cuenta con una librería digital, IEEE Xplore, que provee acceso a través de internet a más de cuatro millones de documentos completos entre algunas de las publicaciones más citadas del mundo en ingeniería eléctrica, ciencias de la computación y electrónica. La biblioteca de IEE Xplore abarca más de 195 *journals*, más de 1800 conferencias, más de 6200 estándares técnicos, aproximadamente 2400 libros y más de 425 cursos educativos. Aproximadamente 20000 nuevos documentos son adicionados a IEEE Xplore cada mes. El repositorio de IEEE dispone de 91564 tripletas en formato RDF ("IEEE - The world's largest technical professional organization for the advancement of technology," n.d.).

- Association for Computing Machinery (ACM)

Repositorio que contiene las publicaciones de la ACM, que es una organización

científica que se encarga de promover el avance en la computación, siendo la organización más grande en esta materia. Cuenta con más de 100000 miembros, de los cuales, más de la mitad se encuentran fuera de los Estados Unidos. ACM cuenta con 37 Grupos de Intereses Especiales, que realizan conferencias, *workshops*, e investigaciones, constituyendo oportunidades de *networking* para sus miembros en constante aumento. Adicionalmente, cuenta con más de 860 capítulos profesionales y estudiantiles, que tienen el mismo propósito, pero de una manera local para establecerse como sistemas de *networking* académico, profesional y un medio de acceso a investigaciones.

La librería digital de ACM (DL) es una base de datos de artículos y literatura bibliográfica relacionadas a computación y tecnologías de la información. Incluye la colección completa de publicaciones de ACM más una base de datos bibliográfica de trabajos principales en computación de editores académicos. La DL se encuentra diseñada para facilitar la diseminación de información, compartirla, interoperabilidad, diseño centrado en el usuario y colaboración entre profesionales en computación, investigadores y educadores. Cuenta con 12402336 tripletas en formato RDF ("Association for Computing Machinery," n.d.).

4.2.2. Pruebas

Se procedió a consultar los SPARQL Endpoints de los Datasets y sus propiedades relevantes escogidas. Se construyó la consulta mediante el API de Jena, una consulta para cada palabra seleccionada mediante el módulo de procesamiento de lenguaje del objetivo anterior. Una vez realizado esto, se ordenaron todos los resultados, según su relevancia y se construyó la estructura RDF inicial.

Los repositorios científicos mencionados anteriormente cuentan con un SPARQL Endpoint, que permite realizar consultas en SPARQL, dichos Endpoints son los siguientes:

DBLP: http://dblp.rkbexplorer.com/sparql/

IEEE: http://ieee.rkbexplorer.com/sparql/

ACM: http://acm.rkbexplorer.com/sparql/

```
PREFIX id:   <http://ieee.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX akts: <http://www.aktors.org/ontology/support#>
PREFIX iai:  <http://www.iai.uni-sb.de/resist#>

select * where {?s ?p ?o . FILTER  (regex(?s, 'machine, 'i')  || regex(?o, 'machine', 'i'))}
LIMIT 100
```

Ilustración 7. SPARQL query para cada palabra de la consulta

Se realizó una consulta por cada palabra a cada Endpoint, como se puede apreciar en la imagen 7. Primero se especifican los prefijos por defecto de cada Endpoint, en el caso del ejemplo corresponden a los de IEEE, luego se elabora propiamente la consulta, en la cual se aplicó un filtro por similitud al sujeto y objeto del triple, sin especificar ninguna condición para el predicado, se muestra cada resultado completo y se limita la ejecución a los primeros 100 resultados, para evaluarlos y escoger los más adecuados.

Una vez obtenidos los resultados, se procedió a agruparlos, priorizando aquellos recursos que coinciden para todas las palabras de la consulta. Para esto se utilizó una estructura con la clase de Java HashMap, que permite almacenar un valor con un identificador único como se muestra en el procedimiento del pseudocódigo 1.

```
procedimiento OrdenarPorRelevancia
1: list <- resultados de consulta SPARQL
2: map <- { }
3: para cada statement en list:
4:        buscar statement en map
5:        si statement ∈ map
6:                map[statement] <- map[statement] + 1
7:        fin si
8:        si statement ∉ map
9:                map[statement] <- 1
10:       fin si
11: fin para
12: ordenar map en orden decreciente
13: fin procedimiento
```

Pseudocódigo 1. Procedimiento Ordenar por Relevancia

Se creó una estructura HashMap cuyos identificadores correspondían a los valores sin duplicados de los statements de la lista de resultados. Si no existe el identificador, se crea uno con el valor del contador en 1, y si existe, se incrementa dicho valor de tal manera que, al ordenarlos de manera decreciente, los primeros resultados corresponden de manera más precisa a los recursos que se buscaron con la cadena de búsqueda.

Utilizando esta estructura, se plasmó el modelo de tripletas RDF en una base de datos relacional para poder procesarla en la interfaz y poder obtener información relevante, como estadísticas de las propiedades más frecuentes, entre otras. El modelo de base de datos se puede observar en la imagen 8.

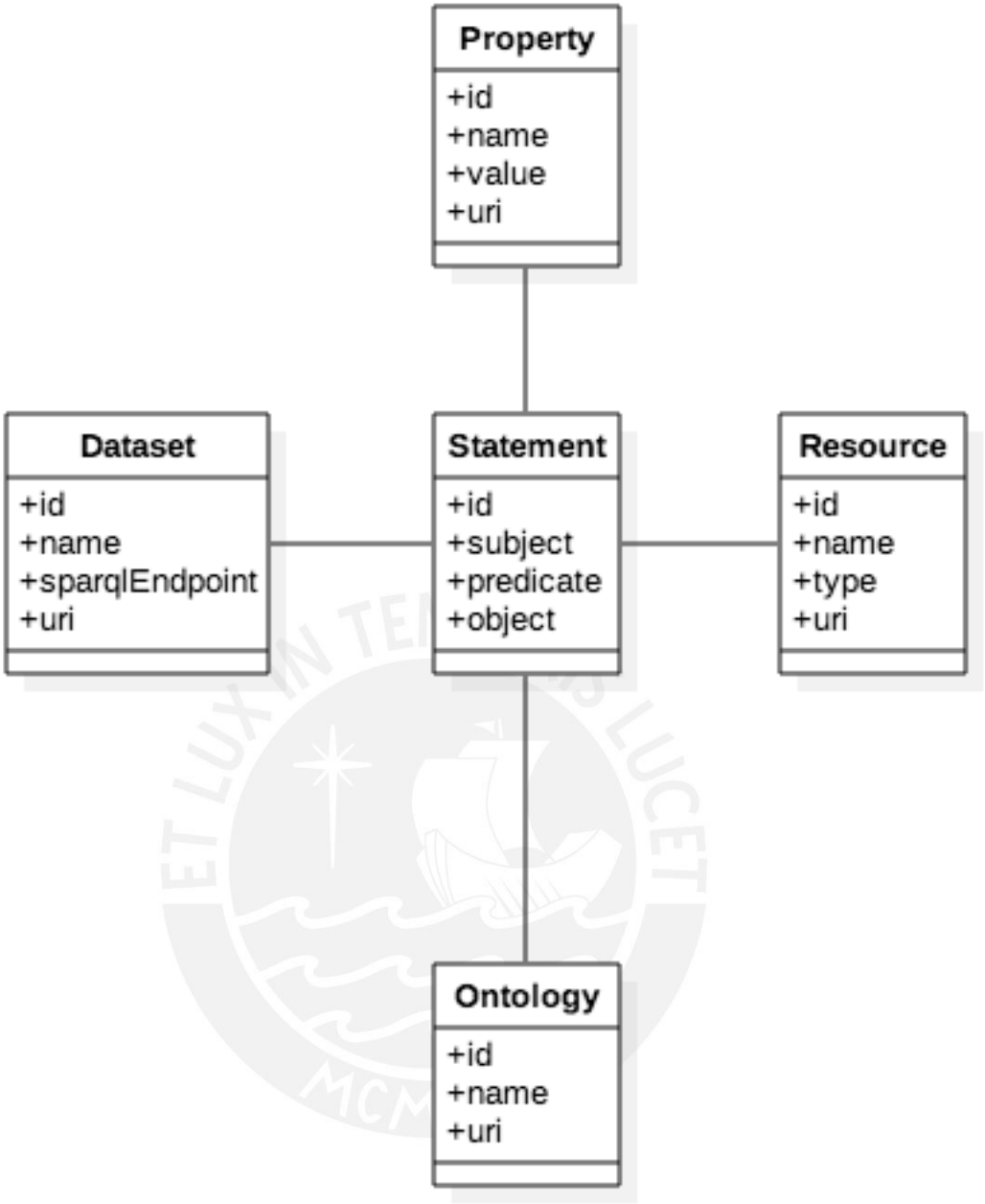

Ilustración 8. Modelo de base de datos

URI del recurso	**Valor del recurso**
http://dblp.rkbexplorer.com/id/conf/ifip/KilburnGS59	Experiments in machine learning and thinking.
http://dblp.rkbexplorer.com/id/journals/ai/Griffith74	A Comparison and Evaluation of Three Machine Learning Procedures as Applied to the Game of Checkers.

http://dblp.rkbexplorer.com/id/conf/icga/Englander85	Machine Learning of Visual Recognition Using Genetic Algorithms.
http://dblp.rkbexplorer.com/id/conf/sigir/WongZ86	A Machine Learning Approach to Information Retrieval.
http://dblp.rkbexplorer.com/id/journals/ml/Langley86a	Editorial: The Terminology of Machine Learning.
http://ieee.rkbexplorer.com/id/publication-00848452	The technique employs a machine learning method, inductive logic programming (ILP), for synthesizing first order logic formulas that describe the valid operations of a program from the normal runs of the program[…]
http://acm.rkbexplorer.com/id/980011	Machine learning and vectorial matching for an image retrieval model: EXPRIM and the system RIVAGE
http://acm.rkbexplorer.com/id/1033486	A Machine Learning Approach to Improve Congestion Control over Wireless Computer Networks
http://acm.rkbexplorer.com/id/855124	An Error Reducing Approach to Machine Learning using Multi-Valued Functional Decomposition
http://acm.rkbexplorer.com/id/980980	Machine learning methods applied to DNA microarray data can improve the diagnosis of cancer

Tabla 6. Resultados de la consulta procesada

En base a la consulta definida en el objetivo anterior, se construyó y ejecutó el SPARQL Query, cuyos resultados se pueden apreciar en la Tabla 6, se muestra la URI del recurso, que sería su identificador único en la Web Semántica, junto al valor obtenido. Se limitó la consulta a 10 resultados para comprobar el contenido de la información y que se obtiene de las fuentes esperadas, como se observa en las URIs de los recursos. Las propiedades obtenidas corresponden a títulos, resúmenes y palabras claves o conceptos definidos en las publicaciones obtenidas.

Capítulo 5. OBJETIVO ESPECÍFICO 3: DISEÑAR E IMPLEMENTAR UN MECANISMO QUE PERMITA EXTRAER INFORMACIÓN DE UNA ESTRUCTURA RDF.

5.1. Introducción

La consulta inicial del usuario será asociada a un concepto contenido en la estructura RDF obtenida, sin embargo, hasta este punto no es posible el descubrimiento de nueva información, una premisa bajo la cual fue diseñada la Web Semántica. Para esto se aprovechará la estructura del modelo RDF, basado en las relaciones que poseen los conceptos bajo el modelo de un grafo; mediante estas relaciones se obtendrán los conceptos asociados al primer concepto obtenido, contribuyendo así a brindarle información relevante y permitiendo el descubrimiento de información al usuario.

5.2. Resultado Esperado 3: Mecanismo que permita obtener información relacionada entre trabajos académicos

En este resultado se implementará el mecanismo que permita obtener la información relacionada a la obtenida inicialmente, dicha información será extraída del grafo RDF creado en el objetivo anterior y, mediante las relaciones encontradas en el grafo y el lenguaje de consulta SPARQL, se llegará a obtener, procesar y presentar la información deseada.

5.2.1. Resultado alcanzado

En el grafo RDF obtenido, la información relacionada a la consulta será definida como los trabajos relacionados a las publicaciones obtenidas a través de la consulta inicial. Obtener esta información será posible gracias a que cuentan con propiedades que indican dichas relaciones. La ontología IAI (Institute of Artificial Intelligence) Ontology cuenta con propiedades que relacionan el contenido de ciertas publicaciones; estas propiedades son:

iai:is-related-to

iai:is-strongly-related-to

iai:is-very-strongly-related-to

Estas 3 propiedades, según el modelo RDF tienen en sus asertos como sujeto y objeto las URIs de los trabajos referenciados que comparten alguna relación. Para relacionar el valor de la propiedad obtenida al valor de alguna propiedad del trabajo relacionado se utilizará otra SPARQL Query, que tomará como referencia la URI del recurso obtenido y la lista de propiedades mencionadas.

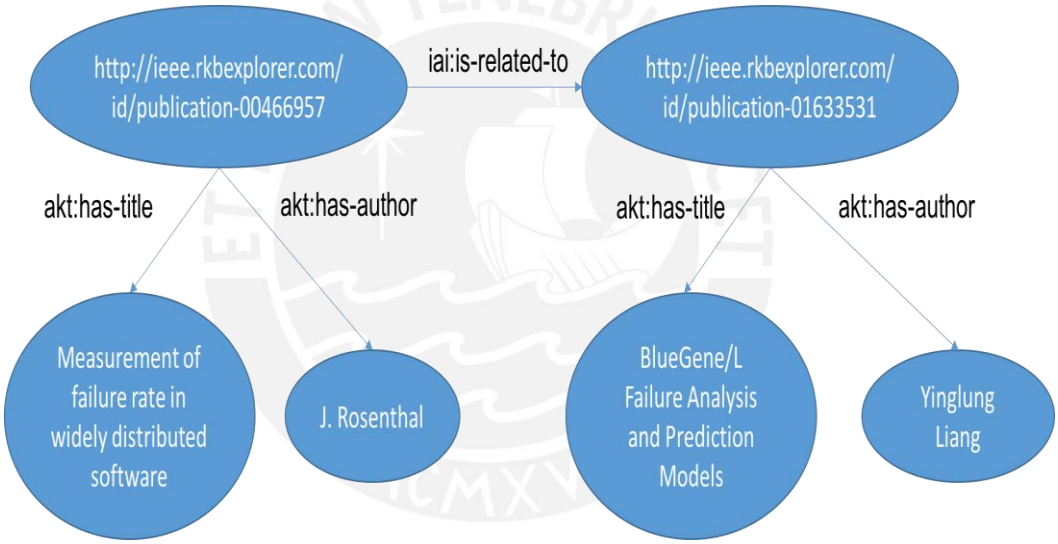

Ilustración 9. Diagrama de la propiedad de relación entre recursos

Las relaciones del grafo RDF se pueden apreciar en la imagen 9; se muestran 2 URIs, que representan el recurso que referencia a 2 publicaciones, ambas con propiedades ejemplo que las identifiquen, la primera es el sujeto de una propiedad, iai:is-related-to que tiene como objeto a la segunda publicación, indicando que existe una relación entre ambas.

Al obtener la estructura RDF inicial, la información se encuentra asociada a la primera URI, mostrando las propiedades relevantes que esta posee, como el título de la publicación, palabras clave o su resumen; si se detecta que dicha URI posee alguna de las propiedades que la relaciona con alguna otra, se

navegará a través de dicha relación, llegando al recurso referenciado y obteniendo las propiedades de este nuevo recurso, que le permitirá al usuario obtener información relacionada a la búsqueda inicial.

5.2.2. Pruebas

Inicialmente se utilizó la información obtenida del grafo RDF inicial, pero dichos recursos no contaban con valores en las propiedades que los relacionaban a otras publicaciones en los Datasets escogidos; por esta razón se decidió mostrar diferentes valores que permitan apreciar la relación entre la información que se puede obtener entre recursos asociados. Para esto, se construyó un nuevo SPARQL Query que consultó el endpoint de la IEEE, buscando relaciones entre las propiedades mencionadas anteriormente, cuya estructura se puede apreciar en la imagen 10.

```
PREFIX id:   <http://ieee.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX akts: <http://www.aktors.org/ontology/support#>
PREFIX iai:  <http://www.iai.uni-sb.de/resist#>

select ?uriS ?titleS ?uriO ?titleO ?urlS ?urlO where { ?uriS iai:is-related-to ?uriO . ?uriS
akt:has-title ?titleS . ?uriO akt:has-title ?titleO . ?uriS akt:has-web-address ?urlS . ?uriO
akt:has-web-address ?urlO . FILTER  (regex(?titleS, '"+searchTerm+"', 'i'))} LIMIT 10
```

Ilustración 10. SPARQL query para las relaciones entre propiedades

La consulta devolverá un conjunto en donde cada resultado se compone de 2 tripletas, uno corresponde a una publicación y el otro a su publicación relacionada por cualquiera de las 3 propiedades, de igual manera limitándose a 10 resultados. ‘searchTerm’ es la palabra clave que indicará una búsqueda por similitud que contenga dicha cadena de texto.

URI del recurso inicial	Valor del recurso inicial	URI del recurso referenciado	Valor del recurso referenciado
http://ieee.rkbexplorer.com/id/publication-5906bbaab75b70d34e85e800c02f940b	Panel on Dependability Benchmarking Overview: Methods, Techniques and Approaches	http://ieee.rkbexplorer.com/id/publication-01311962	Workshop on Architecting Dependable Systems (WADS)
http://ieee.rkbexplorer.com/id/publication-96c24936bbac206a8b406ee9ef48e6bb	Workshop on Dependability Benchmarking	http://ieee.rkbexplorer.com/id/publication-01311962	Workshop on Architecting Dependable Systems (WADS)
http://ieee.rkbexplorer.com/id/publication-01311956	On benchmarking the dependability of automotive engine control applications	http://ieee.rkbexplorer.com/id/publication-01311962	Workshop on Architecting Dependable Systems (WADS)
http://ieee.rkbexplorer.com/id/publication-b5f0d1ccaa12664f9590dd7179ea419f	Software Dependability of a Telephone Switching System	http://ieee.rkbexplorer.com/id/publication-01633542	Cost-Effective Configuration of Content Resiliency Services Under Correlated Failures
http://ieee.rkbexplorer.com/id/publication-00005289	PODS revisited-a study of software failure behaviour	http://ieee.rkbexplorer.com/id/publication-01633542	Cost-Effective Configuration of Content Resiliency Services Under Correlated Failures
http://ieee.rkbexpl	Fault-Tolerance	http://ieee.rkbe	Cost-Effective

orer.com/id/publication-a9782f88662402cc0991374fecde2c0d	in the Advanced Automation System (Invited)	xplorer.com/id/publication-01633542	Configuration of Content Resiliency Services Under Correlated Failures
http://ieee.rkbexplorer.com/id/publication-1a7ad34522dcecd5aae62ecd1f10441e	Fault-Locating Test Generation for Combinational Logic Networks	http://ieee.rkbexplorer.com/id/publication-01633531	BlueGene/L Failure Analysis and Prediction Models
http://ieee.rkbexplorer.com/id/publication-a9782f88662402cc0991374fecde2c0d	Fault-Tolerance in the Advanced Automation System (Invited)	http://ieee.rkbexplorer.com/id/publication-01633531	BlueGene/L Failure Analysis and Prediction Models
http://ieee.rkbexplorer.com/id/publication-00089362	Anomaly detection for diagnosis	http://ieee.rkbexplorer.com/id/publication-01633531	BlueGene/L Failure Analysis and Prediction Models
http://ieee.rkbexplorer.com/id/publication-b82f497583ed258a9b29685087926239	Design of Fault-Tolerant Clocks with Realistic Failure Assumptions	http://ieee.rkbexplorer.com/id/publication-01633531	BlueGene/L Failure Analysis and Prediction Models

Tabla 7. Relación entre recursos académicos

La Tabla 7 muestra la relación entre los recursos consultados; se muestran la URI del recurso sujeto con el valor de la propiedad del título y la URI del recurso objeto, también con el valor de la propiedad del título de la publicación a la que referencia el recurso. Como se puede apreciar, para un mismo recurso, existen más de una referencia, por ejemplo, las tres primeras entradas de la tabla tienen como recurso objeto la publicación titulada “Workshop on Architecting Dependable Systems (WADS)”, que tiene referencias a tres publicaciones distintas, las que serán presentadas finalmente al usuario como información

relacionada a la consulta inicial.

5.3. Resultado Esperado 4: Estructura RDF desreferenciable que permita la navegación entre conceptos

En el resultado anterior se creó un grafo RDF que contiene toda la información asociada a los Datasets de repositorios científicos; dicha información será procesada para encontrar información relevante y permitir el descubrimiento de nueva información, además de servir como conexión con el usuario al material académico, ahora se procederá a obtener información sobre conceptos relacionados a la consulta.

5.3.1. Resultado alcanzado

Para la creación de esta nueva estructura de información, se trabajó con DBPedia, un Dataset que contiene la información de Wikipedia en un formato estructurado, accesible en la Web Semántica. Se toma la misma consulta inicial y se crea un grafo inicial RDF con las propiedades definidas en la ontología de DBPedia; las propiedades escogidas para este resultado permitirán la navegación a través de los recursos de esta estructura.

URI de la propiedad	**Valor de la propiedad**
http://dbpedia.org/ontology/abstract	dbo:abstract
http://dbpedia.org/ontology/subject	dct:subject
https://www.w3.org/2000/01/rdf-schema#comment	rdfs:comment
https://www.w3.org/2000/01/rdf-schema#label	rdfs:label
https://www.w3.org/2000/01/rdf-schema#seeAlso	rdfs:seeAlso
http://dbpedia.org/ontology/ academicDiscipline	dbo:academicDiscipline
http://dbpedia.org/ontology/nonFictionSubject	dbo:nonFictionSubject

https://www.w3.org/2009/08/skos-reference/skos.html#broader	skos:broader
http://dbpedia.org/ontology/ discipline	dbp:discipline

Tabla 8. Propiedades seleccionadas para visualizar conceptos

Las propiedades seleccionadas para este resultado pueden visualizarse en la Tabla 8, estas propiedades fueron elegidas de tal manera que un recurso referenciado por el objeto de un triple obtenido permita la navegación, no son propiedades sujetas al dominio, lo que permitió navegar entre conceptos así estos no se encuentren estrechamente relacionados. Estas propiedades se encuentran almacenadas en la base de datos del proyecto, permitiendo así modificar, agregar nuevas o eliminar propiedades con el fin de mejorar los resultados obtenidos en base a estas propiedades.

5.3.2. Pruebas

Se inició con la obtención de una estructura RDF inicial, en base a esta se obtuvo la información que se muestra en la tabla 9, que corresponde a los resultados de buscar en concepto “Computer Science” en DBPedia, se muestran valores en forma de literales o URIs que permiten continuar la navegación.

Propiedad	**Valor**
dbo:abstract	The computer sciences are those that cover the theoretical bases of information and computation, as well as its application in computer systems. The body of knowledge in computer science is often described as the systematic study of algorithmic processes that describe and transform information: its theory, analysis, design, efficiency, implementation and application.
dct:subject	dbc:Computer_engineering
rdfs:seeAlso	dbr:Computer_programming
dbo:academicDis	dbr:International_Journal_of_High_Performance_Compu

cipline	ting_Applications
dct:subject	dbc:Electronic_engineering
dct:subject	dbc:Electrical_engineering

Tabla 9. Conceptos obtenidos para la consulta "Computer Science"

Para explorar la navegación, se obtuvo la información de alguna de las propiedades, en este caso, dbc:Computer_engineering, que devolvió una estructura referenciando únicamente URIs esta vez como se puede apreciar en la tabla 10.

Propiedad	**Valor**
skos:broader	dbc:Computing
skos:broader	dbc:Engineering_disciplines
skos:broader	dbc:Computer_systems
dct:subject	dbr:Computer_engineering
dct:subject	dbr:Processor_design

Tabla 10. Conceptos obtenidos a partir de la propiedad dbc:Computer_engineering

Se decidió probar un nivel más, obteniendo esta vez el valor de la propiedad dbc:Computer_systems, mostrando una estructura similar al grafo RDF obtenido anteriormente, como se puede observar en la Tabla 11. Estas propiedades brindaran al usuario la información sobre los conceptos asociados a la búsqueda y que guardan una relación indicada en cada propiedad.

Propiedad	**Valor**
dct:subject	dbr:Fault_tolerance
dct:subject	dbr:Integration_appliance
skos:broader	dbc:Technology_systems
skos:broader	dbc:Computer_engineering

Tabla 11. Conceptos obtenidos a partir de la propiedad dbc:Computer_systems

Capítulo 6. OBJETIVO ESPECÍFICO 4: DISEÑAR E IMPLEMENTAR UN COMPONENTE QUE PERMITA LA VISUALIZACIÓN DE LA INFORMACIÓN.

6.1. Introducción

Una vez obtenida toda la información necesaria y relevante sobre el concepto inicial y los relacionados de la Web Semántica se necesita de una componente visual en la forma de una interfaz gráfica de usuario, para que este pueda interactuar sin necesidad de conocimientos técnicos sobre las tecnologías y procesos detrás de la obtención de información de una manera fácil e intuitiva.

6.2. Resultado Esperado 5: Navegador con una interfaz gráfica de usuario intuitiva.

Para la implementación de este navegador se utilizará el lenguaje de programación Java, mediante el framework Spring, junto con el entorno de desarrollo Eclipse, que brinda plugins para utilizar distintas herramientas que faciliten el trabajo del desarrollador. El software tendrá una arquitectura web, detallada en la siguiente sección, de tal manera que el usuario pueda acceder desde cualquier entorno con acceso a internet.

6.2.1. Resultado alcanzado

La interfaz consiste en una vista principal con un menú lateral que contiene las vistas de la solución.

La primera vista corresponde al proceso principal, es decir, la búsqueda, compuesto únicamente por una barra para ingresar la cadena de consulta y un botón para proceder con la búsqueda. La pantalla de los resultados se mostrará de la siguiente manera:

La primera es la sección de conceptos, en donde se visualizarán los valores de las propiedades previamente definidas en la ontología, a manera de una tabla.

En esta sección el usuario podrá navegar a través de las propiedades que tengan relaciones dentro del grafo RDF obtenido internamente.

La segunda sección se centrará en mostrar la información obtenida de los repositorios académicos, mostrando las propiedades de las publicaciones, así como trabajos relacionados con sus respectivos enlaces al sitio original de las publicaciones, permitiendo así el descubrimiento de información.

La siguiente vista corresponde a la búsqueda avanzada, en donde el usuario podrá personalizar la búsqueda a través de filtros que especifican los resultados de la búsqueda. Estos filtros se encuentran agrupados en función de la vista de resultados de búsqueda, en donde cada grupo cuenta con las propiedades de los Datasets que se manejan en cada sección, filtrando que propiedades se desean mostrar o especificando valores para estas.

La arquitectura de la solución se presenta como una serie de vistas según lo especificado en el modelo 4+1, que se pueden apreciar en la Imagen 11: vista de despliegue, lógica, de procesos, física y de casos de uso (Kruchten, 1995).

Ilustración 11. Diagrama del Modelo 4 + 1

- Vista Lógica

  Separa los componentes de la aplicación en agrupaciones lógicas, compuesta de 5 paquetes o capas principales, cuyo diagrama se ilustra en la Imagen 12.

  - Modelo: Representa la información con la cual el sistema funcionará y gestionará el acceso a la misma, también podrá conectarse con la base de datos y poder guardar nueva información. Se utilizará el ORM Hibernate incluida con el Framework Spring.
  - Controlador: Se encarga de procesar las peticiones solicitadas en la vista y entregarlas al modelo, el cual contiene la lógica del negocio.
  - Vista: Presenta el Modelo de una manera gráfica, con las clases visuales que crean las ventanas con las que el cliente interactúa con la solución, en este caso será utilizada la tecnología JSP para manejar los archivos de la vista
  - Librerías: Las librerías que se trabajadas se actualizarán usando el administrador de dependencias web Maven.

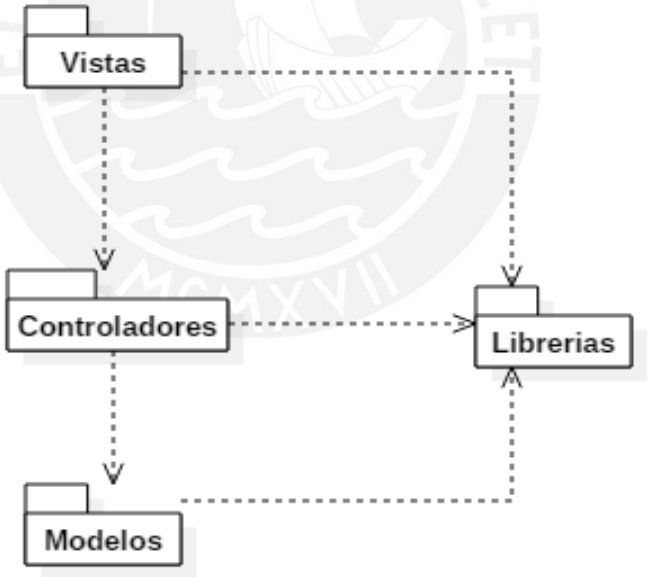

Ilustración 12. Diagrama de la vista lógica

- Vista de Proceso

  Aquí se presentan los flujos de trabajo de los componentes que conforman la aplicación, que se muestran en la Imagen 13:

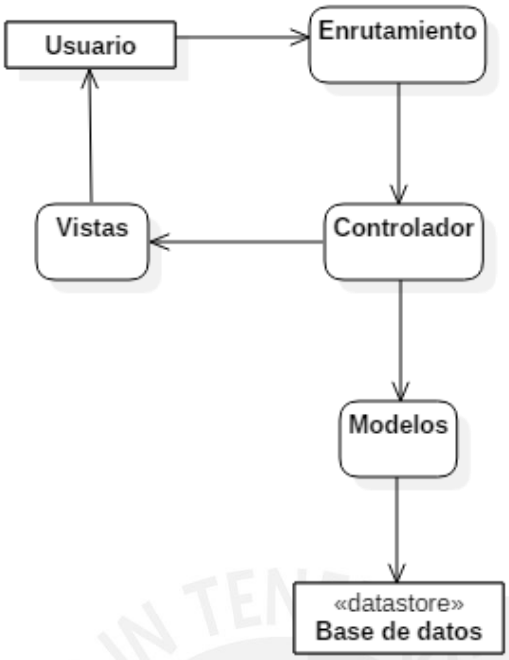

Ilustración 13. Diagrama de actividades - Vista de Proceso

- Vista de Despliegue

  Muestra la distribución física de los componentes físicos de la aplicación y su interacción, como se puede apreciar en la Imagen 14:

  - Cliente Web: Representa el equipo del usuario que use la aplicación desde su computadora. Dentro se encuentran los componentes de interfaz gráfica y muestra una relación con la lógica del negocio de la aplicación.
  - Servidor Web: Dentro se encuentran dos nodos: el de servidor de aplicaciones y el servidor de base de datos.
  - Servidor de aplicaciones: Dentro se encuentran los componentes de la lógica del negocio y del acceso a los datos, en este caso se utilizará Apache como servidor web.
  - Servidor de base de datos: Dentro se encuentra físicamente la base de datos, en este caso se utilizará MySQL.

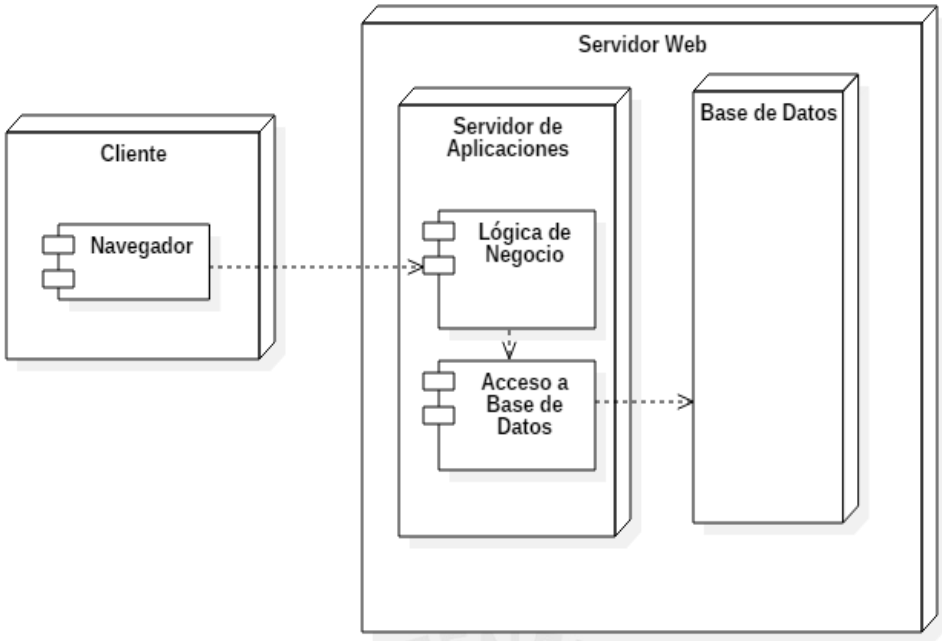

Ilustración 14. Diagrama de Despliegue

- Vista de Implementación

  La aplicación cuenta con seis capas: la capa de Enrutamientos, la capa de Controladores, la capa de View, la capa de Servicios, la capa de Modelos, y la capa Hibernate las cuales se muestran en la Imagen 15:

  - Enrutamientos: Esta capa procesa la solicitud HTTP generado por el navegador con una ruta específica e indica que controlador debe responder.
  - Controladores: Esta capa se encarga de recibir datos de la solicitud HTTP y poder validarlos, llamar a la capa de Servicios, así como también enviar datos a la capa de Vistas.
  - Servicios: Esta capa encierra las reglas de negocio de la aplicación. Se comunicará con los Modelos para poder representar un caso de uso.
  - Modelos: Representan los objetos de negocio, así como también se encargarán de su persistencia en la base de datos, ya que implementan el patrón Active Record, es decir la conexión entre las clases con sus respectivas tablas en base de datos con una configuración mínima.
  - Vistas: Recibe los datos del Controlador una vez que ya se comunicó con los Modelos para obtener la información que se debe mostrar en el navegador al usuario

- Hibernate: Esta capa contiene el ORM Hibernate utilizado para el mapeo de la base de datos utilizada con el sistema.

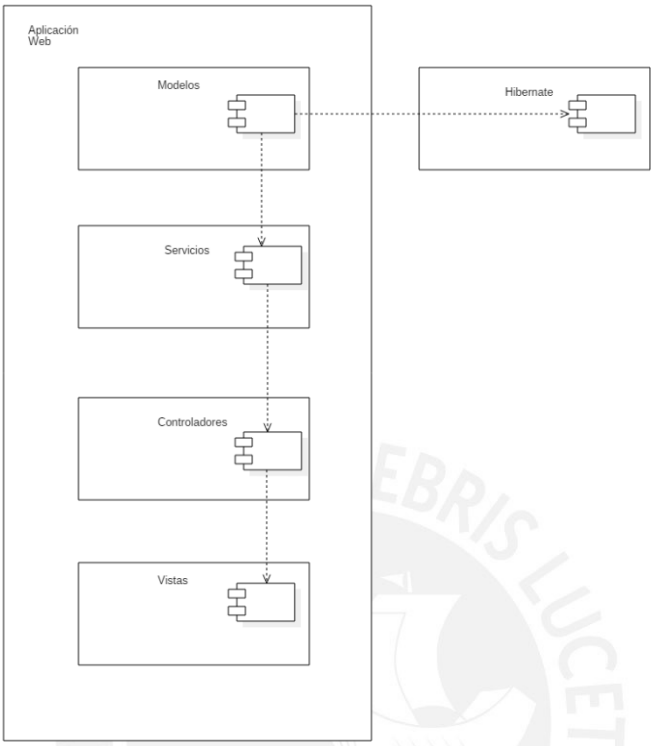

Ilustración 15. Vista de Implemetación

6.2.2. Pruebas

Se realizó una prueba de la funcionalidad del navegador desde el inicio, probando con la consulta “Computer Science”, para poder obtener los conceptos relacionados y la información relevante extraída de los Datasets de los trabajos científicos

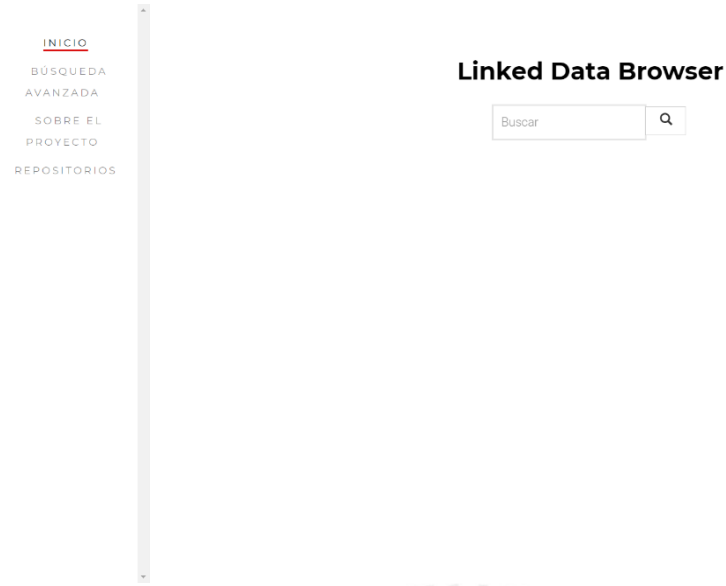

Ilustración 16. Pantalla de búsqueda inicial

En la Imagen 16 se muestra la interfaz de la pantalla de búsqueda; es una interfaz minimalista formada por una entrada de texto para ingresar la cadena y un botón para iniciar la búsqueda. Esta vista fue elaborada con la intención de que el usuario pueda utilizarla sin dificultad alguna y de manera intuitiva.

A medida que el usuario tipea la cadena, aparece una opción para autocompletar la barra de búsqueda con términos relacionados. Dichos términos fueron extraídos a partir de propiedades de DBPedia relacionadas a ciencias de la computación, de tal manera que si el usuario elige alguna de ellas es una búsqueda que garantizará resultados y a su vez facilitará el proceso de búsqueda.

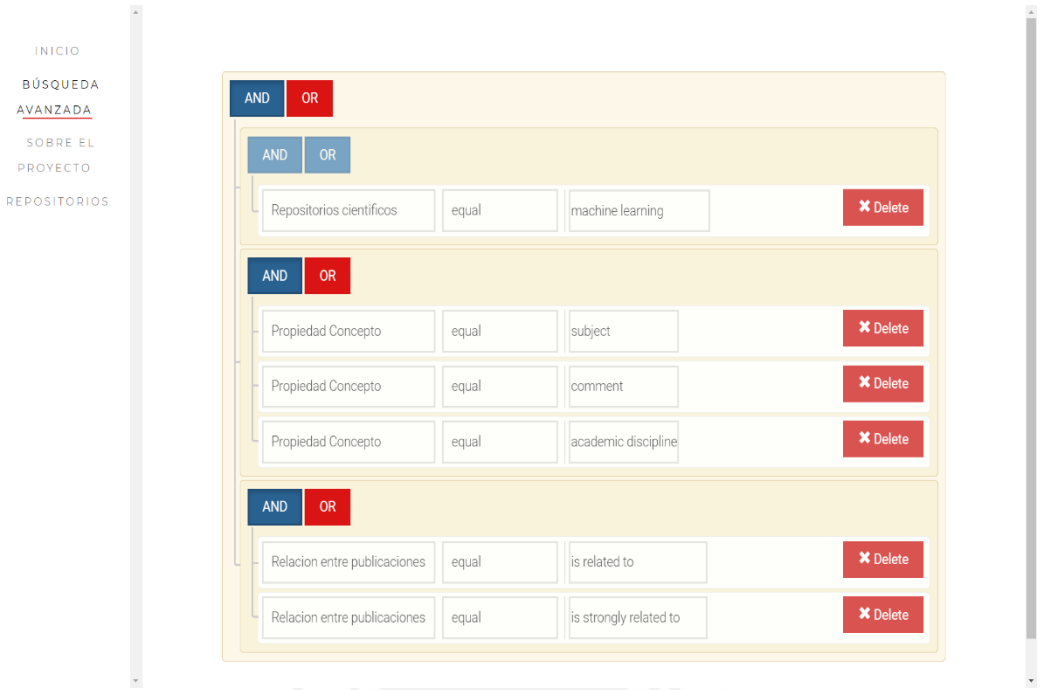

Ilustración 17. Pantalla de búsqueda avanzada

En la Imagen 17 se muestra la vista de la búsqueda avanzada, que se encuentra formada por 3 grupos de filtros. El primero permite una búsqueda por coincidencia similar a la búsqueda general, el segundo permite seleccionar propiedades específicas de DBPedia de las seleccionadas para el proyecto y el último corresponde a la sección de relaciones entre publicaciones científicas, permitiendo especificar el grado de la relación. Una vez seleccionados todos los filtros se procede a la búsqueda que llevará a la misma vista de resultados que la búsqueda general.

Ilustración 18. Pantalla de conceptos navegables

Los resultados de la cadena de búsqueda ingresada se muestran en la imagen 18, que corresponde a la primera sección de la vista de los resultados. Esta sección contiene la lista de conceptos obtenidos del Dataset de DBPedia, diferenciando valores literales de URIs que permiten la navegación en el grafo RDF para seguir explorando conceptos relacionados. Si el usuario ingresa a alguno de estos recursos, se mostrará una tabla con el mismo formato, indicando las propiedades y los valores para el nuevo concepto escogido. De esta manera se puede navegar tan profundamente como se desee, siempre y cuando los conceptos no escapen del dominio de Ciencias de la Computación.

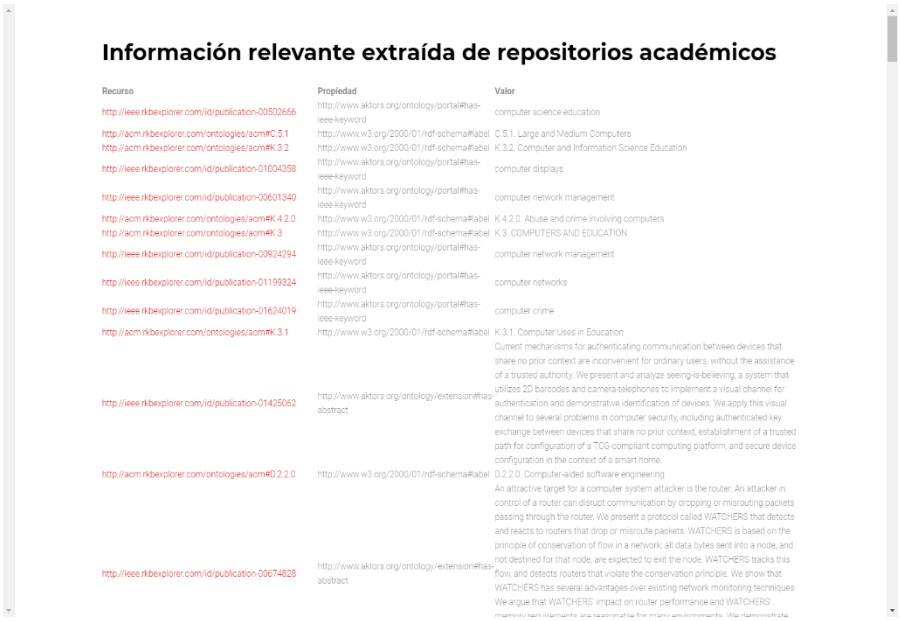

Ilustración 19. Pantalla de resultados de fuentes académicas

La segunda parte de los resultados se muestra en la imagen 19, la cual contiene los resultados de la información relevante extraída de los Datasets de los repositorios académicos, mostrando una tabla con la URI del recurso, que corresponde a la publicación, el valor de la propiedad y el valor del objeto de la tripleta, que corresponde a la información que se desea visualizar.

Ilustración 20. Pantalla de relaciones entre publicaciones

Finalmente, la última sección se muestra en la imagen 20, que corresponde a las relaciones entre publicaciones científicas. Se muestra el URI del recurso de la publicación inicial, su título y el URI de la publicación original, en caso de que se encuentre disponible. El URI de la publicación va a tener una referencia en caso de que se encuentren publicaciones asociadas a esta, mediante las relaciones que lo especifican en el grafo RDF, si el usuario le da click a una de ellas, en navegador mostrará el detalle de estas relaciones en una tabla con el título de las publicaciones con sus referencias al documento original, brindándole acceso al usuario para que pueda obtener toda esta información. De esta manera se permite el descubrimiento de información; el usuario tenía una consulta inicial, el navegador brinda conceptos en base al procesamiento de la consulta, brinda información académica y finalmente información relacionada a la consulta inicial, que probablemente el usuario desconocía y que será relevante para él.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1. Conclusiones

La Web Semántica y la nube de Linked Open Data son una fuente valiosa de información, estructurada para el entendimiento tanto de personas como de máquinas y pueden cambiar la forma en la que funciona la Web actualmente, dado que la Web Semántica es en sí una extensión de la Web tradicional y fue creada inicialmente para soportar este formato. Pero para que esto sea posible, se necesita que toda la información actual sea estructurada bajo los estándares de RDF y Linked Data. Este proyecto contribuye a dicho propósito, brindando una herramienta que consulta información en Datasets asociados al dominio de Ciencias de la Computación.

La información contenida en los Datasets y las herramientas actuales que brindan acceso a estos, como los SPARQL Endpoints, por lo menos en el dominio de ciencias de la computación, se encuentran hasta cierto punto desorganizadas, no existe un estándar universal para la definición de propiedades; consultas SPARQL que funcionan en un determinado Endpoint, generan errores en otro, dificultando la navegación entre distintos repositorios, debido a que tiene que elaborar una consulta SPARQL distinta para cada una. Esto puede ser una razón por la cual las tecnologías de la Web Semántica no se encuentran tan difundidas actualmente.

Si bien estas tecnologías presentan numerosos beneficios, una contraparte es la dificultad técnica a la hora de utilizarlas; si un usuario desea consultar información en la nube de Linked Open Data, los Datasets existentes brindan SPARQL Endpoints para que los usuarios ingresen consultas, que requieren un dominio del lenguaje SPARQL. Por otra parte, la información se encuentra estructurada bajo el modelo RDF, cuyas especificaciones frecuentemente se encuentran documentadas bajo el estándar OWL, el cual no es entendible si no se cuentan con los conocimientos necesarios de dichas tecnologías. Es por estas razones que se buscó implementar una herramienta que no requiera los conocimientos en las tecnologías y herramientas mencionadas y, a su vez, sea

sencillo e intuitivo de utilizar, brindándole al usuario la oportunidad de acceder a la información contenida en la Web Semántica.

El proyecto se enfocó en brindarle al usuario información tanto sobre conceptos navegables como información extraída de repositorios académicos que permitan el descubrimiento de información, lográndolo gracias a la estructura de los Datasets consultados, cuyo modelo cuenta con propiedades que al ser aprovechadas mediante el modelo RDF y el lenguaje de consulta SPARQL permiten encontrar relaciones, en diferentes magnitudes, entre publicaciones científicas.

7.2. Recomendaciones y trabajos futuros

Este Proyecto fue implementado para facilitar la búsqueda y descubrimiento de información relevante, pero se limita al dominio de Ciencias de la Computación, por lo que sería posible incluir dominios adicionales en el proyecto, separándolos posiblemente como categorías de búsqueda distintas para que los resultados sean únicamente del dominio deseado, permitiendo así expandir el alcance del proyecto no solamente a investigadores especializados en ciencias de la computación, sino también a especialistas de cualquier área, dependiendo de los dominios que se agreguen.

Adicionalmente, con respecto al mecanismo utilizado para procesar la consulta, este utiliza técnicas de procesamiento de lenguaje natural que se limitan a la transformación de la cadena de búsqueda con la intención de obtener posibles conceptos, así como una búsqueda por similitud entre términos similares para minimizar una búsqueda sin resultados en caso de haber cometido errores en la elaboración de la cadena de búsqueda. Sin embargo, se pueden aplicar técnicas adicionales relacionadas a un análisis semántico de la consulta, permitiendo así la creación de consultas más elaboradas, por ejemplo, en formato de preguntas u oraciones negativas del tipo ¿Qué no es ingeniería de software?

El buscador cuenta con una opción de autocompletado, que fue implementada

con la intención de facilitar el proceso de búsqueda al usuario. Esta opción muestra sugerencias obtenidas de un conjunto de propiedades del Dataset de DBPedia, que podría no contener el contenido de la cadena que el usuario desea. Una forma de mejorar esto puede ser incluyendo más propiedades de Datasets adicionales y si la memoria para almacenarlos no es suficiente, implementar un mecanismo que cargue sugerencias a medida que se va tipeando la consulta, en lugar de cargar todas las sugerencias a la vez.

Por otra parte, los repositorios de información están compuestos de bibliotecas de tripletas RDF en inglés, dado que es el idioma que predomina en la Web Semántica, razón por la cual se implementó la solución en dicho idioma. Para incluir un idioma adicional como característica, se pueden incluir Datasets en los idiomas deseados, pero debido a la escasez de estos, sería más factible implementar un mecanismo de traducción que procese la información de los Datasets consultados actualmente y a partir de esta, traduzca los resultados para que la solución tenga un mayor alcance.

Referencias Bibliográficas

Alfonso, F., Bermejo, J., & Segovia, J. (2005). Duplicate or redundant publication: can we afford it? *Revista Española de Cardiología (English Edition)*, *58*(5), 601–604.

[9] "Apache Jena." . [En línea]. Disponible en: https://jena.apache.org/. [Consultado: 20-mar-2017].

Association for Computing Machinery. (n.d.). Retrieved from https://www.acm.org/

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722–735). Springer.

Average typing speed. (n.d.). Retrieved from https://www.ratatype.com/learn/average-typing-speed/

Batzios, A., & Mitkas, P. A. (2012). WebOWL: A Semantic Web search engine development experiment. *Expert Systems with Applications*, *39*(5), 5052–5060.

Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., … Sheets, D. (2006). Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd international semantic web user interaction workshop* (Vol. 2006, p. 159). Citeseer.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, *284*(5), 1–5.

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205–227.

DBLP - Computer Science Bibliography. (n.d.). Retrieved from http://dblp.uni-trier.de/

[10] "Eclipse - The Eclipse Foundation open source community website." . [En línea]. Disponible en: https://eclipse.org/. [Consultado: 20-mar-2017].

Genesereth, M. R., & Nilsson, N. J. (1987). Logical foundations of artificial. *Intelligence. Morgan Kaufmann*, *58*.

Heeringa, W. J. (2004). *Measuring dialect pronunciation differences using Levenshtein distance*. Citeseer.

IEEE - The world's largest technical professional organization for the advancement of technology. (n.d.). Retrieved from https://www.ieee.org/index.html

Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., … Pollack, M. (2004). The spring framework–reference documentation. *Interface*, *21*.

Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn.

Kim, Y., Yoo, S., & Park, S. (2012). A semantic Web browser for novice users. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on* (pp. 806–809). IEEE.

Kruchten, P. B. (1995). The 4+ 1 view model of architecture. *IEEE Software*, *12*(6), 42–50.

[15] "Linked Data - Design Issues." . [En línea]. Disponible en: https://www.w3.org/DesignIssues/LinkedData.html. [Consultado: 29-abr-2016].

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)* (pp. 55–60).

Meeker, M. (2015). Internet trends 2015-Code conference. *Glokalde*, *1*(3).

[8] "OWL Web Ontology Language Overview." . [En línea]. Disponible en: https://www.w3.org/TR/2004/REC-owl-features-20040210/. [Consultado: 29-abr-2016].

Peirce, C. S., Hartshorne, C., & Weiss, P. (1932). Collected Papers of Charles Sanders Peirce. Vol. I, Principles of Philosophy.

[14] "Query - W3C." . [En línea]. Disponible en:

https://www.w3.org/standards/semanticweb/query. [Consultado: 29-abr-2016].

Rational Software. (1998). Rational Unified Process. Best Practices for Software Development Company.

[5] “RDF - Semantic Web Standards.” . [En línea]. Disponible en: https://www.w3.org/RDF/. [Consultado: 29-abr-2016].

[12] “Semantic Web roadmap.” . [En línea]. Disponible en: https://www.w3.org/standards/semanticweb/data. [Consultado: 05-may-2016].

Shen, Z., Hou, Y., Li, C., & Li, J. (2012). Voovle: A linked data search engine for scientific data. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on* (pp. 1171–1175). IEEE.

[6] “SPARQL Query Language for RDF.” . [En línea]. Disponible en: https://www.w3.org/TR/rdf-sparql-query/. [Consultado: 29-abr-2016]

[7] “TaskForces/CommunityProjects/LinkingOpenData/DataSets.” . [En línea]. Disponible en: https://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets. [Consultado: 20-mar-2017]

Toutanova, K., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13* (pp. 63–70). Association for Computational Linguistics.

[13] “W3C Semantic Web Activity.” . [En línea]. Disponible en: https://www.w3.org/2001/12/semweb-fin/w3csw. [Consultado: 29-abr-2016].

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

IMPLEMENTACIÓN DE UN NAVEGADOR DE CONCEPTOS ENLAZADOS EN EL DOMINIO DE CIENCIAS DE LA COMPUTACIÓN

Tesis para optar por el **Título de Ingeniero Informático**, que presenta el bachiller:

Alexis Enrique León Shimabukuro

Asesor: Hector Andrés Melgar Sasieta

Lima, marzo del 2017

INDICE

1. ANEXO 1: Especificación de etiquetas utilizadas por el POS Tagger de la Universidad de Stanford

Abreviatura	Significado
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POST	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present

VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

Tabla 1. Especificación de las etiquetas del POS Tagger

2. ANEXO 2: Diagrama de clases del modelo de base de datos

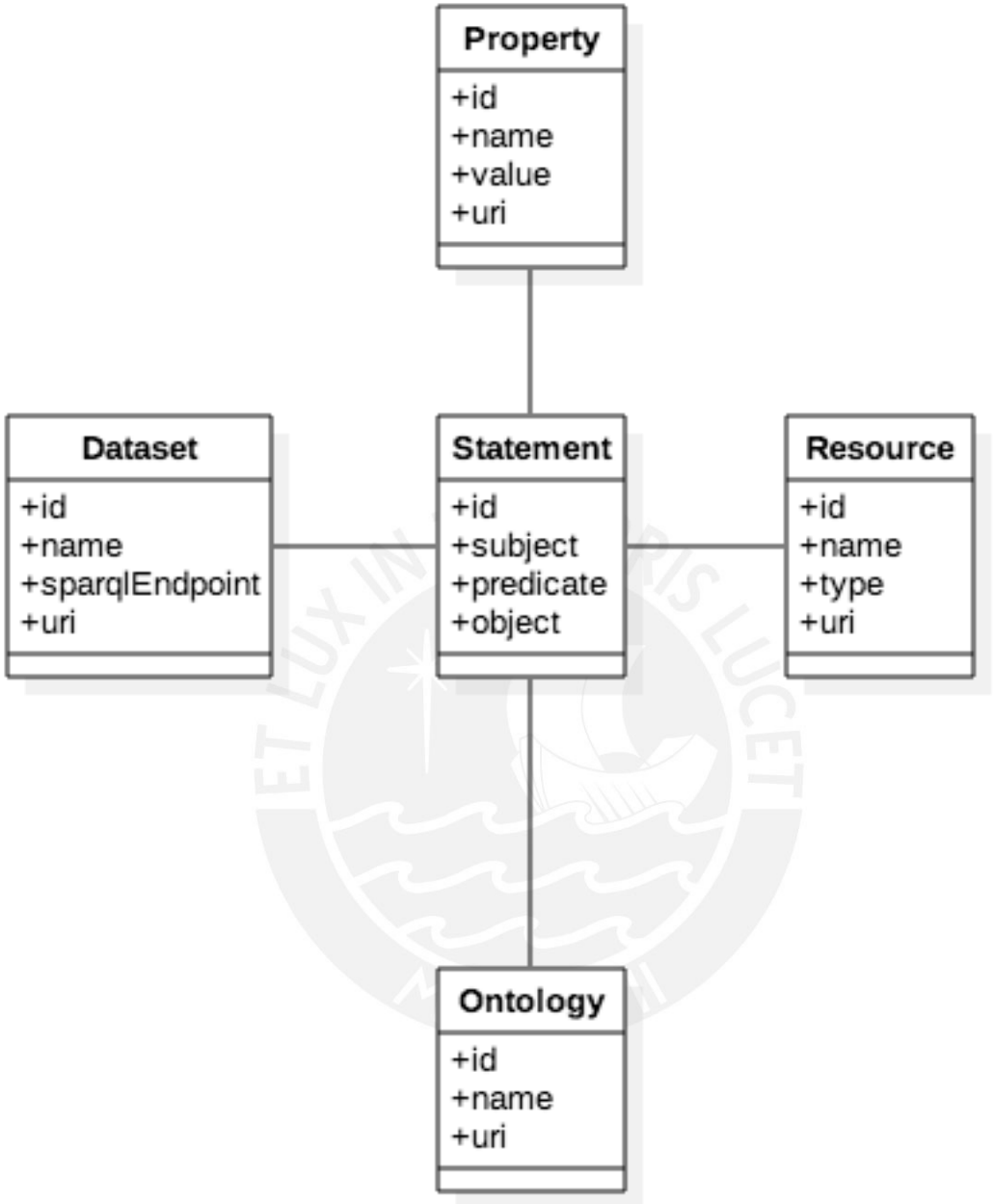

Ilustración 1. Diagrama de clases del proyecto