



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

Grupo de Investigación de Web Semántica

Componente para la transformación, enlazado y publicación de grafos RDF

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor:

Luis Manuel Pérez Batista

Tutores:

MSc. Yusniel Hidalgo Delgado

Ing. Ernesto Ortiz Muñoz

La Habana, junio de 2016

“Año 58 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Luis Manuel Pérez Batista
Autor

MSc. Yusniel Hidalgo Delgado
Tutor

Ing. Ernesto Ortiz Muñoz
Tutor

DATOS DE CONTACTO

Síntesis del Tutor

El ingeniero Yusniel Hidalgo Delgado se graduó con Título de Oro en la Universidad de las Ciencias Informáticas en el año 2010. En su primer año de adiestramiento desempeñó diversos roles dentro del proyecto de desarrollo del ERP cubano. Actualmente se desempeña como profesor asistente del departamento docente de técnicas de programación de la Facultad 3. Es coordinador del grupo de investigación de Web Semántica de la UCI. Es miembro de la Asociación Cubana de Reconocimiento de Patrones, de la Sociedad Cubana de Matemática y Computación y de la *International Association for Pattern Recognition*.

DEDICATORIA

A Osmaida y Salvador, mis padres y mejores maestros.

A mi familia y amigos.

A mi tutor y profesores.

A mis colegas del grupo de Web Semántica.

A Cuba.

AGRADECIMIENTOS

Gracias a mi familia, en particular a Osmaida y Salvador, por la formación, el amor y el apoyo que me han brindado en todos estos años.

Gracias a Margarita, por su amor, por siempre estar ahí, en las buenas y en las malas.

Gracias a los amigos que siempre han estado a mi lado.

Gracias a los profesores que contribuyeron a mi formación.

Gracias a mis colegas del movimiento de programación competitiva Tomás López Jiménez, por ayudarme a mejorar mis habilidades como programador.

Gracias a mi tutor, por sus conocimientos y por adentrarme en el camino de la investigación.

Gracias al grupo de Web Semántica, por sus valiosas recomendaciones.

A todos, muchas gracias.

RESUMEN

Las actividades de transformación, enlazado y publicación de datos enlazados se encargan de convertir los datos provenientes de diversas fuentes hacia el modelo de datos RDF. Estas actividades permiten generar un grafo RDF donde la información se encuentre preparada para ser procesada por las tecnologías de la web semántica. En los últimos años se han realizado investigaciones con el objetivo de mejorar las aproximaciones existentes siguiendo el principio de los datos enlazados. El grupo de Web Semántica de la Universidad de las Ciencias Informáticas se encuentra desarrollando un proyecto donde se aplican entre otras actividades las previamente mencionadas. En el presente trabajo se propone un componente para la transformación, enlazado y publicación de grafos RDF a partir de bases de datos relacionales. Con la implementación del componente se pretende disminuir el tiempo empleado en la ejecución de estas actividades.

Palabras claves: datos enlazados; enlazado; grafo RDF; publicación; transformación

ABSTRACT

The transforming, linking and publishing linked data activities are responsible for converting data from several sources to the RDF data model. These activities allow generating an RDF graph where the information is ready to be processed by the Semantic Web technologies. In recent years, research has been conducted with the goal of improving existing approaches following the linked data principles. The Semantic Web group of the University of Information Science is developing a project where, among other activities apply the mentioned before. In this work, a component for transforming, linking and publishing RDF graphs from relational databases is proposed. With the implementation of the component, it is to reduce the time spent in the execution of these activities.

Keywords: *linked data; linking; publishing; RDF graph; transforming*

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	8
1.1. Introducción	8
1.2. Marco teórico	8
1.3. Estado del arte	10
1.3.1. Enfoques para la transformación.....	12
1.3.2. Enfoques para el enlazado.....	16
1.3.3. Enfoques para la publicación	19
1.4. Conclusiones parciales	21
CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA	23
2.1. Introducción	23
2.2. Descripción general de la propuesta	23
2.3. Metodología de desarrollo de software.....	23
2.4. Tecnologías empleadas	23
2.5. Técnicas de captura de requisitos	24
2.6. Arquitectura del componente.....	24
2.6.1. Transformación	25
2.6.2. Enlazado.....	27
2.6.3. Publicación.....	28
2.7. Estándares de codificación.....	29
2.8. Requisitos funcionales de software	30
2.9. Historias de usuario.....	30
2.10. Requisitos no funcionales	35
2.10.1. Software.....	36
2.10.2. Hardware	36
2.10.3. Diseño.....	37
2.11. Conclusiones parciales.....	37
CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA	38
3.1. Introducción	38
3.2. Planificación de pruebas	38
3.2.1. Pruebas internas	38
3.2.2. Pruebas de aceptación.....	38
3.3. Pruebas de software	39
3.3.1. Pruebas de caja blanca.....	39
3.3.2. Resultados de las pruebas unitarias.....	42

ÍNDICE

3.3.3. Pruebas de caja negra	43
3.3.4. Resultados	49
3.4. Caso de estudio	50
3.4.1. Descripción	50
3.5. Análisis de los resultados	51
3.6. Conclusiones.....	52
CONCLUSIONES GENERALES	53
RECOMENDACIONES	54

ÍNDICE DE TABLAS

Tabla I: Comparativa de guías metodológicas para los proyectos de datos enlazados	11
Tabla II: Comparativa de las herramientas estudiadas para la transformación	16
Tabla III: Comparativa de las herramientas estudiadas para el enlazado	18
Tabla IV: Comparativa de las herramientas estudiadas para la publicación	21
Tabla V: Requisitos funcionales	30
Tabla VI: Historia de Usuario Detectar cambios en la BDR	30
Tabla VII: Historia de Usuario Transformar grafo RDF	31
Tabla VIII: Historia de Usuario Enlazar grafo RDF	31
Tabla IX: Historia de Usuario Editar archivo de alineación	32
Tabla X: Historia de Usuario Editar archivo de transformación	33
Tabla XI: Historia de Usuario Editar archivo de enlazado	34
Tabla XII: Historia de Usuario Editar archivo de publicación	35
Tabla XIII: Caso de prueba para el camino básico #1	41
Tabla XIV: Caso de prueba para el camino básico #2	41
Tabla XV: Caso de prueba para el camino básico #3	42
Tabla XVI: Caso de prueba de aceptación CP-01	43
Tabla XVII: Caso de prueba de aceptación CPA-02	45
Tabla XVIII: Caso de prueba de aceptación CPA-03	45
Tabla XIX: Caso de prueba de aceptación CPA-04	46
Tabla XX: Caso de prueba de aceptación CPA-05	46
Tabla XXI: Caso de prueba de aceptación CPA-06	48
Tabla XXII: Caso de prueba de aceptación CPA-07	49
Tabla XXIII: Propuesta para el diseño experimental.	51
Tabla XXIV: Análisis de los resultados.	51

ÍNDICE DE FIGURAS

Fig. 1: Modelo de datos basado en grafo del estándar RDF. (Fuente: elaboración propia).....	9
Fig. 2: Guías metodológicas propuestas a) Fuente: Hidalgo Delgado 2015 , b) Fuente: Gómez-Pérez et al. 2013	11
Fig. 3: Arquitectura del proyecto Biblioteca Digital Semántica	24
Fig. 4: Arquitectura del componente Publicación de datos enlazados	25
Fig. 5: Método para la transformación de grafos RDF. Fuente: elaboración propia.	26
Fig. 6: Fragmento de archivo de alineación mapping.ttl. Fuente: elaboración propia.	26
Fig. 7: Función que obtiene las diez primeras tripletas del grafo RDF	40
Fig. 8: Grafo de flujo asociado al método obtenerDiezPrimerasTripletasRdf	40
Fig. 9: Resultados de las pruebas unitarias en la primera iteración	43
Fig. 10: Resultados de las pruebas unitarias en la segunda iteración.....	43
Fig. 11: Resultados de las pruebas de aceptación	50
Fig. 12: Interfaz gráfica de la vista Alineación.....	59
Fig. 13: Interfaz gráfica de la vista Transformación	59
Fig. 14: Interfaz gráfica de la vista Enlazado	60
Fig. 15: Interfaz gráfica de la vista Publicación	60

INTRODUCCIÓN

Desde sus inicios, la web ha representado un paso de avance en el desarrollo tecnológico de la humanidad. El salto de complejidad que media desde la concepción de la web como un conjunto de documentos HTML enlazados entre sí por hipervínculos, hasta la aparición de las primeras aplicaciones web o la construcción de las redes sociales disponibles hoy en día, ha supuesto un desafío constante para las personas involucradas en el desarrollo de las tecnologías web que han hecho posible tal evolución (Gómez-Pérez et al. 2013). La web se ha convertido en el principal medio de comunicación para las personas y el medio ideal para la adquisición de información. Con la primera versión de la web, la Web 1.0, solo se concebían páginas web estáticas, vinculadas entre ellas y editadas manualmente por desarrolladores web. Estos vínculos establecían las relaciones entre los datos de manera que adquirieran significado para las personas, no siendo así para las máquinas, las cuales no podían procesar los datos de las páginas ni de los recursos almacenados en la web (Allemang y Hendler 2011).

Con la llegada de la Web 2.0 o Web Clásica, se da un paso de avance en lo que respecta a la interacción humano-máquina, siendo posible por cualquier usuario insertar, modificar o eliminar datos sin poseer privilegios administrativos sobre los sitios web. Se profundizan en la sociedad algunas actividades como el comercio electrónico, las transacciones bancarias, los cursos en línea con una extensa variedad de contenidos didácticos, la prensa digital, tanto local, nacional e internacional, el intercambio cultural, apoyándose en los foros y redes sociales, la consulta de información a través de los motores de búsqueda, entre otras actividades que les permiten a los usuarios ser protagonistas del desarrollo de la web.

Aunque la web actual revolucionó el modo en que se desarrollaban las actividades socioeconómicas y la forma de gestionar la información, aún posee un conjunto de limitaciones que le impiden aprovechar todas sus potencialidades (Murugesan 2007). Las principales limitaciones son el formato, la integración y la recuperación de los recursos disponibles en la web. En la web actual, la mayoría de los recursos web se encuentran en el formato *Hypertext Markup Language* (HTML), siendo este comprendido por los navegadores para darle formato a los contenidos orientados a los usuarios. No obstante, estos contenidos no pueden ser procesados por las computadoras, por lo cual no es posible extraer su valor semántico (García y Delgado 2015). La dispersión de los datos, trae consigo otro problema debido a que no existe una relación explícita entre los mismos, dificultando así su utilización por sistemas informáticos. El formato y la dispersión evitan una efectiva recuperación de la información, evidenciándose en los motores de búsquedas actuales, los cuales en la mayoría de los casos, no satisfacen totalmente a los criterios de consultas formulados por los usuarios.

Para solucionar las limitaciones de la web anteriormente mencionadas, surge la Web Semántica, Web de los Datos o Web 3.0. Su creador, Tim Berners-Lee, la define como: “... *La Web Semántica no pretende sustituir la web actual, sino que es una extensión de la misma en la que la información tiene un significado bien definido, posibilitando a los humanos y las computadoras trabajar en cooperación*” (Berners-Lee et al. 2001). En este sentido, la *World Wide Web Consortium (W3C)*, de conjunto con investigadores de todo el mundo, han trabajado en la última década en la definición de varios estándares, muchos de los cuales han sido utilizados en el desarrollo de múltiples aplicaciones (Hidalgo Delgado y Rodríguez Puente 2013). La Web Semántica tiene como objetivo fundamental que las páginas web no sólo las entiendan las personas sino que también puedan ser usadas como fuente de conocimiento por sistemas informáticos (Criado Fernández 2009). Para que la web de datos sea efectiva se necesita un lenguaje común que permita especificar los recursos existentes en la web y las distintas relaciones que existen entre ellos, así como poder consultar estos datos mediante aplicaciones computacionales.

Para llevar a cabo este nuevo escenario, el W3C ha definido estándares que contribuyen a disminuir el efecto que provocan las limitaciones de la web actual y a mejorar la estructuración e integración de los contenidos en la web. En este contexto, Tim Berners-Lee enuncia el concepto de datos enlazados como: “... *conjunto de buenas prácticas para la publicación y enlazado de datos estructurados en la Web*” (Berners-Lee 2006). Los datos enlazados son la base del modelo empleado por la Web Semántica donde cada recurso, dígame una página web, un archivo digital o un concepto abstracto, posee un *Uniform Resource Identifier (URI)* que lo diferencia del resto y permite el enlazado con otros recursos.

Un URI posee un nombre, un protocolo, una autoridad y un camino hacia el recurso que el mismo describe (Berners-Lee et al. 1994; GUTIÉRREZ 2007). Un *Uniform Resource Locator (URL)* es un URI que, además de identificar un recurso en la web, permite actuar sobre dicho recurso describiendo un método de acceso hacia su localización. Por su parte, el modelo de datos *Resource Description Framework (RDF)* es el estándar del W3C para describir información semántica sobre recursos en la web. Este modelo está basado en el uso de grafos dirigidos y etiquetados compuesto por tripletas del tipo sujeto – predicado – objeto, siendo el sujeto y el objeto los nodos del grafo y el predicado la arista que une a dichos nodos (Berners-Lee et al. 2001). La capacidad que posee este modelo de datos para procesar metadatos facilita la interoperabilidad entre diversas aplicaciones, proporcionando un mecanismo para el intercambio de información a través de la web.

Como se afirma en la recomendación W3C, RDF tiene distintas áreas de aplicación; como la recuperación de recursos (proporcionando mejores prestaciones a los motores de búsqueda), la catalogación en bibliotecas digitales (especificando también las relaciones de contenido disponibles en un sitio web determinado), los agentes inteligentes (facilitando el intercambio de conocimiento), en sistemas de gestión de propiedad intelectual (expresando políticas de privacidad de un determinado objeto) (Adida et al. 2008).

Para la descripción semántica de los contenidos en la web, la W3C propone el uso de ontologías, definiéndose así las especificaciones OWL DL (Criado Fernández 2009). Según (García y Delgado 2015) una ontología es una especificación explícita y formal de una conceptualización compartida. Es explícita porque es descrita en términos de un lenguaje, formal ya que es comprensible por una máquina, es además una conceptualización compartida ya que es una forma de describir y entender un dominio de acuerdo al consenso entre un grupo o varias partes. Las ontologías definen conceptos y relaciones de algún dominio, de forma compartida y consensuada. El lenguaje OWL permite representar el conocimiento acerca de conceptos bien definidos y las relaciones entre ellos. Este es un lenguaje basado en lógica computacional que permite el razonamiento mediante programas de computadoras que verifican su estructura e infieren el conocimiento implícito a partir del conocimiento explícito (Hidalgo Delgado y Rodríguez Puente 2013). El uso de ontologías permite crear una estructura de datos sólida para especificar las reglas existentes entre los diferentes conceptos y permitir que los datos tengan un valor semántico.

Otro aspecto importante para el desarrollo de la web semántica lo constituye el empleo de guías metodológicas para la publicación de datos enlazados. Existen muchas guías metodológicas que comparten procesos comunes, pero todas siguen un método iterativo incremental que cubre los procesos comunes de extracción, transformación, enlazado, publicación y consumo de datos enlazados. El primer proceso en común es el de la extracción de los datos, el cual contiene actividades importantes como el análisis y selección de las fuentes de datos, la generación de los URIs y la definición de ontologías.

La transformación de los datos tiene una estrecha dependencia con la extracción debido a que la mayoría de los datos que existen en internet se encuentra contenida en bases de datos relacionales. Investigaciones recientes demuestran que es posible trasladar de bases de datos relacionales a RDF empleando para ello lenguajes de alineación, los cuales se encargan de sincronizar los datos según su tipología. Una vez almacenadas todas las tripletas en el grafo RDF se procede al enlazado. El enlazado se encarga de unir las tripletas RDF que se relacionen según su descripción o las ontologías previamente declaradas. Este enlace se establece a través de las URIs entre tripletas y un grafo RDF

o entre dos grafos (Gómez-Pérez et al. 2013). En la publicación juegan un papel importante la asignación de tres tipos de URIs, una que identifica el origen del recurso, otra que muestra la descripción del recurso como RDF y una que proporciona una vista de la descripción del recurso y sus relaciones en forma de una página web. El consumo garantiza que los datos puedan ser procesados por programas capaces de utilizar el estándar RDF. Estos programas suelen ser buscadores semánticos, razonadores y los encargados de enlazar datos en RDF.

La progresiva penetración de las nuevas tecnologías de la información y la comunicación (TIC) en las bibliotecas, la universalización del uso de Internet y la diversificación de los recursos que se pueden hacer accesibles desde la red, han provocado que las bibliotecas se involucren en un proceso de reinención que implica una profunda revisión de sus técnicas y sus metodologías de trabajo y de los servicios que prestan para, de esta forma, poder satisfacer mejor las demandas cada vez más exigentes y específicas de sus usuarios (Cabrera Facundo y Coutín Domínguez 2005; Marshall 1997; Saracevic 2000). Como fruto de este proceso de transformación y adaptación, surgen las bibliotecas digitales, una extensión lógica de las bibliotecas físicas que alberga colecciones de recursos en formato electrónico (bien en su origen, bien tras ser sometidos a procesos de digitalización) y que tienen asociadas una serie de servicios para facilitar el acceso a estos recursos a diferentes comunidades de usuarios utilizando para ello diversas tecnologías (Giannaccini y Müller 2015).

Las bibliotecas digitales son uno de los principales nodos de acceso a la información en la Web. El proyecto de datos enlazados en bibliotecas consiste en la transformación y publicación de los catálogos bibliográficos en RDF. Estas actividades se realizan utilizando las ontologías o vocabularios estándares de la Federación Internacional de Asociaciones de Bibliotecas (IFLA), para que los datos sean accesibles en la web semántica y enlazarlos con otros conjuntos de datos de la nube como el Fichero de Autoridades Virtual Internacional (VIAF), y en un futuro próximo, con DBpedia. Para representar los recursos en RDF, se han utilizado ontologías o modelos consensuados por la comunidad de bibliotecas y propuestos por IFLA. El portal datos.bne.es representa una de las primeras iniciativas a nivel internacional en hacer un uso intensivo de los modelos propuestos por IFLA, como el estándar internacional para descripción bibliográfica ISBD, o el modelo de requisitos funcionales para registros bibliográficos FRBR, cuyo propósito es proporcionar una forma estandarizada de describir datos bibliográficos que permita el intercambio de registros a nivel internacional (Vila-Suero, Villazón-Terrazas y Gómez-Pérez 2013).

El grupo de investigación de web semántica de la Universidad de las Ciencias Informáticas está desarrollando un proyecto de investigación que requiere seguir una guía metodológica para el

desarrollo de una biblioteca digital semántica. Para el desarrollo de dicha biblioteca se llevan a cabo los procesos de extracción, limpieza, publicación y consumo de datos enlazados. La presente investigación se enmarca en el proceso de publicación de datos enlazados a partir de bases de datos relacionales. En este proceso se realizan las actividades de transformación, enlazado y publicación de grafos RDF. Sin embargo, las herramientas que se utilizan para realizar estas actividades están totalmente aisladas, sin una correcta integración de las mismas. Esto trae consigo un aumento en el tiempo que se requiere para ejecutar estas actividades, debido a que en estas actividades se ven involucrados diversos estándares y herramientas.

Atendiendo a la problemática anterior, se propone el siguiente **problema a resolver**: ¿Cómo integrar herramientas y estándares para realizar las actividades de transformación, enlazado y publicación de grafos RDF en el contexto de los datos enlazados para disminuir el tiempo de ejecución de estas actividades?

El **objeto de estudio** en el que está centrado la investigación son los datos enlazados enmarcándose en el **campo de acción** de las actividades de transformación, enlazado y publicación de grafos RDF. Como **objetivo general** se plantea desarrollar un componente de software que integre las herramientas y estándares existentes para realizar las actividades de transformación, enlazado y publicación de grafos RDF, disminuyendo el tiempo de ejecución de estas actividades.

A partir de lo planteado anteriormente se definen los siguientes **objetivos específicos**:

1. Elaborar el marco teórico y el estado del arte del objeto de estudio de la investigación mediante el análisis bibliográfico documental para identificar tendencias y adoptar posiciones al respecto.
2. Diseñar un componente de software para la transformación, enlazado y publicación de grafos RDF.
3. Implementar un componente de software para la transformación, enlazado y publicación de grafos RDF.
4. Validar los resultados obtenidos con la utilización del componente de software desarrollado mediante la realización de un diseño experimental.

Atendiendo al problema de la investigación se tiene como **idea a defender**:

Si se desarrolla un componente de software que integre las herramientas y estándares existentes para realizar los procesos de transformación, enlazado y publicación de grafos RDF se disminuirá el tiempo empleado en su utilización.

De lo anterior se obtiene la **variable independiente**: desarrollar un componente de software y la **variable dependiente**: disminución del tiempo de ejecución de estas actividades.

Métodos teóricos

Para el logro de los objetivos de la investigación, se aplicaron varios métodos de investigación científica. A continuación, se detallan cada uno de ellos.

El método **Analítico-Sintético** permitió realizar un análisis sobre la teoría y las tendencias actuales de los procesos de transformación, enlazado y publicación llevados a cabo durante el empleo de las guías metodológicas. Se ha podido estudiar las diferentes metodologías, tecnologías y estándares para la correcta integración de estos procesos previamente mencionados.

El método **Inductivo-Deductivo** ha permitido generalizar los procesos comunes que describen las diferentes guías metodológicas estudiadas. A partir de los conocimientos adquiridos se ha podido generar una propuesta de solución que incluya una correcta integración de los procesos analizados a través del razonamiento lógico.

Métodos empíricos

Dentro de los métodos empíricos utilizados están la **observación** y la **medición**. La observación permitió obtener información relacionada con los estándares y herramientas que han sido utilizados en proyectos similares al que se propone en esta memoria de tesis. La medición se utilizó en la validación de la solución propuesta.

La investigación está estructurada en tres capítulos.

Capítulo 1: Se analizan los principales conceptos relacionados con el trabajo, enmarcados en el contexto de la Web Semántica. Se analizan las diferentes aproximaciones relacionadas con el tema de la investigación. Se analizan las diferentes guías metodológicas enfatizando en las actividades de transformación, enlazado y publicación de grafos RDF, así como las diferentes herramientas utilizadas para su ejecución. Se realiza un estudio de la literatura para identificar los elementos que forman parte de la propuesta de solución y su impacto social.

Capítulo 2: En este capítulo se detalla el sistema desarrollado. Se generan las historias de usuario y se agrupan por iteraciones teniendo en cuenta el nivel de prioridad asociado. Se analizan las tareas que se realizan durante las actividades de transformación, enlazado y publicación de grafos RDF. Se describen los métodos y herramientas utilizadas en la solución del problema planteado en la investigación.

Capítulo 3: En este capítulo se describen las pruebas de la metodología de desarrollo AUP-UCI. Se presentan las iteraciones concebidas por la planificación, así como las tareas planteadas para la

realización de cada una de las historias de usuario. También se realizan las pruebas de aceptación para ratificar que las historias de usuario han sido implementadas adecuadamente al concluir cada iteración, garantizando la calidad del componente. Finalmente se realiza un pre experimento con el objetivo de medir la relación existente entre las variables que intervienen en la idea a defender.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se hace referencia a los fundamentos básicos para la publicación de datos enlazados, destacando los conceptos y definiciones fundamentales. Se realiza un análisis de la literatura para identificar los elementos que pueden formar parte de la propuesta de solución. Se presentan y comparan las herramientas y estándares que permiten realizar las actividades de transformación, enlazado y publicación de datos enlazados. Finalmente, se identifican conceptos relacionados con la web semántica y que son de vital importancia en el ciclo de vida de un proyecto de datos enlazados.

1.2. Marco teórico

En la web existen datos de todo tipo, tales como: datos de diversos dominios (Google académico), datos médicos (Biblioteca Nacional de Medicina de Estados Unidos), datos gubernamentales (London Datastore) y noticias (Cubadebate). La mayoría de estos datos se encuentran almacenados en Bases de Datos Relacionales (BDR) y son consultados frecuentemente por diferentes aplicaciones que se encargan de procesarlos y utilizarlos según su interés (Morales, Soto y Martínez 2005). Una **base de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su uso posterior. El modelo relacional para la gestión de una base de datos se basa en la lógica de predicados y en la teoría de conjuntos. En este modelo los datos son almacenados en correspondencia con sus relaciones, formando una BDR. El principio básico del modelo relacional es realizar una reducción de la redundancia de los datos (Barzdins y Kirikova 2011).

Actualmente los computadores son capaces de procesar y consultar los datos, pero no de entenderlos. La web semántica es un conjunto de recomendación propuestas por la W3C con el objetivo de que las computadoras sean capaces de entender los datos en la web. El término recomendación se refiere a una descripción formal de una tecnología que debería ser utilizada por todos. El W3C es el organismo regulador de los estándares en la web.

Para la transición hacia la Web Semántica, se requiere que la información este correctamente estructurada e integrada. Por tal motivo, en el 2006, Berners Lee enuncia el término de **datos enlazados**: “Los datos enlazados se refieren a un conjunto de buenas prácticas para la publicación y enlazado de datos estructurados en la Web” (Berners-Lee 2006). Estos datos enlazados provienen de diferentes fuentes de datos y con ellos se pueden compartir a escala global datos estructurados que pueden ser reutilizados por diferentes sistemas. Los cuatro principios de datos enlazados son:

1. Usar URIs como nombres para las cosas.
2. Usar URIs HTTP, de tal manera que las personas puedan ver esos nombres.
3. Cuando alguien ve un URIs, provee información útil usando los estándares (RDF, SPARQL).
4. Incluir enlaces a otras URIs, de tal modo que ellos puedan descubrir más cosas (Hogan et al. 2012).

El modelo de datos **RDF** es la propuesta del W3C para representar información sobre recursos en la web (Duesa et al. 2016; Wood et al. 2014). Este es un modelo de datos estándar y con una semántica formal basado en grafos dirigidos y etiquetados para la publicación y enlazado de datos estructurados en la Web. Su sintaxis está basada en triplas del tipo sujeto-predicado-objeto, donde el sujeto describe un recurso en la web, el objeto puede estar representado por un recurso o un literal y el predicado es la relación que existe entre el sujeto y el objeto. Un **recurso** corresponde a un elemento que se encuentra en la web, tal como: una página web, una persona que usa la web, una película, un libro, entre otros.

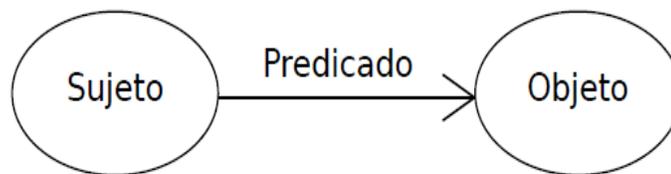


Fig. 1: Modelo de datos basado en grafo del estándar RDF. (Fuente: elaboración propia).

Los grafos RDF están compuestos por al menos una tripleta RDF, las cuales se almacenan secuencialmente en un archivo, una base de datos o un *triplestore* (almacén de tripletas). Como se mencionaba anteriormente, los sujetos y algunos objetos son representados por URIs, lo cual convierte un triple en una especificación con cadenas muy largas, por tal motivo se crean **prefijos** que simplifican las direcciones permitiendo solo almacenar parte de la ruta y referenciar el resto del URI.

El uso de los vocabularios en la web semántica permite la interoperabilidad entre distintos sistemas debido a que ofrecen un marco común para acceder a los recursos y sus propiedades utilizando las mismas palabras. El vocabulario **FOAF** (*Friend Of A Friend*), fue el primer vocabulario común para redes sociales que existió. Este vocabulario integra tres tipos de redes: las redes sociales de colaboración humana, amistad y asociación. Otro término importante para la estructuración de datos almacenados en formato RDF es el de ontología. Una **ontología** es una especificación formal y explícita de una conceptualización compartida. Existen cuatro tipos diferentes de ontologías a diferentes niveles de granularidad:

- Fundacionales: estas ontologías capturan los conceptos generales independientes de un dominio específico.
- Dominio: estas ontologías modelan conceptos y relaciones que son relevantes para un dominio específico. En estas ontologías se suelen reutilizar términos de otras ontologías fundacionales.
- Tareas: las cuales describen conceptos de una tarea en específico.
- Aplicaciones. estas ontologías combinan ontologías de dominio y ontologías de tareas extendiéndolas con nuevos conceptos y relaciones más específicas.

Uno de los lenguajes para representar ontologías en la web es **OWL** (*Web Ontology Language*) (Bechhofer 2009). El mismo está definido sobre el modelo de datos propio de RDF y se encuentra basado en lógica computacional de modo que el conocimiento expresado en OWL puede ser razonado por sistemas informáticos que verifican la consistencia del conocimiento permitiendo que conocimiento implícito se convierta en conocimiento explícito. Define el comportamiento que deben tomar los datos según su clasificación y la sintaxis empleada para su utilización. Los documentos OWL, conocidos como ontologías, pueden ser publicados en la web, los cuales pueden incluso hacer referencia hacia o desde otras ontologías OWL (Garrote Hernández 2014).

El lenguaje de consulta que se utiliza para acceder a los datos almacenados en RDF es **SPARQL**, acrónimo de *SPARQL Protocol and RDF Query Language*. Es un lenguaje declarativo de consultas similar a SQL que permite realizar consultas sobre los datos en un grafo RDF incluyendo entidades provenientes de diferentes servicios y aplicaciones. El resultado de las consultas pueden ser un conjunto de resultados o grafos RDF. También se emplea *SPARQL 1.1 Update* para realizar modificaciones en grafos RDF a través de un *Endpoint Update* (Prud'Hommeaux, Seaborne y others 2008).

1.3. Estado del arte

La Web Semántica no es una web totalmente nueva, sino que es el siguiente paso de la web actual donde se resuelven parte de las limitaciones existentes en la misma. La Web Semántica es soportada sobre la plataforma tecnológica de la web actual manteniéndose el uso del protocolo HTTP para el intercambio de datos. En el contexto de la web, la publicación de datos estructurados siguiendo los principios de los datos enlazados juega un rol importante. En este sentido, diversos autores han propuesto en la literatura un conjunto de guías metodológicas y buenas prácticas para la publicación y consumo de datos enlazados.

El ciclo de vida de un proyecto de datos enlazados se desarrolla llevando a cabo un modelo iterativo incremental. La ejecución de dicho proyecto se realiza teniendo en cuenta las guías metodológicas

para la publicación de datos enlazados, las cuales realizan los procesos que se muestran en la figura 2:

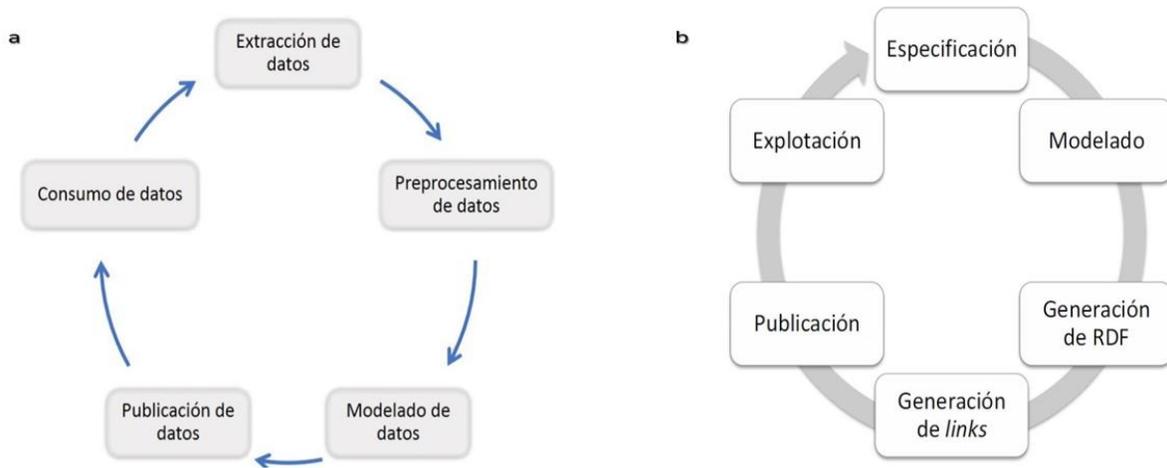


Fig. 2: Guías metodológicas propuestas a) Fuente: Hidalgo Delgado 2015 , b) Fuente: Gómez-Pérez et al. 2013

A continuación, se muestra una tabla comparativa de estas dos guías metodológicas donde se describen las actividades que se realizan en las mismas.

Tabla 1: Comparativa de guías metodológicas para los proyectos de datos enlazados

Comparativa de guías metodológicas	
a	b
Extracción de datos	Especificación
<ul style="list-style-type: none"> Extracción y almacenamiento de los datos provenientes de diferentes orígenes. 	<ul style="list-style-type: none"> Identificación y análisis de los orígenes de datos. Diseño de URIs.
Preprocesamiento de los datos	Modelado
<ul style="list-style-type: none"> Limpeza de los datos. Normalización de los datos. 	<ul style="list-style-type: none"> Buscar vocabularios válidos que se puedan reutilizar y si no existen crearlos. Determinar la ontología que se utilizará para modelar el dominio de los datos originales.
Modelado de datos	Generación
<ul style="list-style-type: none"> El objetivo de esta actividad es determinar la o las ontologías que serán utilizadas en el modelado de los datos. 	<ul style="list-style-type: none"> Transformar los datos al modelo RDF. Limpiar errores e inconsistencia en los datos.

	<ul style="list-style-type: none"> • Crear enlaces entre los nodos del grafo y hacia otros grafos.(Se recomienda utilizar el <i>framework</i> SILK o LIMES)
Publicación de datos	Publicación
<ul style="list-style-type: none"> • Transformar los datos al modelo RDF. • Crear enlaces entre los nodos del grafo y hacia otros grafos. • Publicar los grafos RDF y los metadatos asociados a cada uno de ellos. 	<ul style="list-style-type: none"> • Publicar los grafos RDF y los metadatos asociados a cada uno de ellos. • Habilitar el descubrimiento de los grafos RDF en la web.
Consumo de datos	Explotación
<ul style="list-style-type: none"> • Desarrollar aplicaciones que utilicen los datos enlazados publicados en los grafos RDF (<i>Bibliotecas digitales</i>). 	<ul style="list-style-type: none"> • Desarrollar aplicaciones que utilicen los datos enlazados publicados en los grafos RDF.(<i>Navegadores y motores de búsqueda de datos enlazados</i>)

En la actualidad estas actividades se realizan de manera independiente, empleando métodos y herramientas diferentes para ejecutar cada tarea. De las dos guías metodológicas estudiadas, la variante propuesta por (Hidalgo-Delgado, 2015) se ajusta a las necesidades del proyecto, debido a que integra las actividades de transformación, enlazado y publicación en un solo proceso, mientras que la segunda guía metodológica separa la actividad de publicación en un proceso distinto al enlazado y la transformación. La publicación de datos enlazados tiene como objetivo proveer un método para el descubrimiento de grafos RDF en la web.

1.3.1. Enfoques para la transformación

Esta actividad consiste en la transformación de datos desde las fuentes de datos seleccionadas en la actividad de Extracción usando el modelo ontológico desarrollado en la actividad de Modelado. Esta es una actividad crucial en el proceso de publicación y se encuentra influenciada por la identificación del lenguaje de alineación utilizado en la actividad de transformación (Gómez-Pérez et al. 2013). A continuación, se describen un grupo de enfoques encontrados en la literatura para realizar el proceso de transformación.

1.3.1.1. Virtuoso RDF Views

Virtuoso RDF Views¹ realiza consultas SQL a partir de consultas formuladas en SPARQL. Las consultas se realizan a través de una capa de alineación relacional en un servidor conocido como RDF Views. El término RDF Views se refiere a las dos tecnologías claves que soportan a Virtuoso RDF. La transformación realizada utiliza un metalenguaje declarativo para definir la alineación de datos SQL a ontologías RDF. La alineación se realiza de forma dinámica, en consecuencia, los cambios en los datos subyacentes se reflejan inmediatamente en la representación RDF, sin cambios en el esquema relacional subyacente.

1.3.1.2. D2RQ++

D2RQ++ es una herramienta para la traslación bidireccional de bases de datos relacionales en almacenes RDF virtuales. Esta herramienta es una extensión de D2RQ, la cual solo generaba y mostraba tripletas RDF a partir de BDR. Esta aproximación involucra algoritmos para trasladar y actualizar tripletas RDF en atributos o tuplas equivalentes en un esquema de BDR y extensiones para soportar traslaciones de estructuras de nodos blancos a tuplas relacionales equivalentes (Eisenberg y Kanza 2012).

1.3.1.3. upRDF

La herramienta upRDF se basa en la creación de la estructura adecuada para la detección y obtención de los cambios realizados a partir de una BDR. Se crean funciones que inserten los datos en una tabla nombrada *check_updates* y se crean los *triggers* asociados a cada tabla que invocan a las correspondientes funciones. Para la generación de las tripletas RDF se necesita el documento de alineación (*mapping.ttl*). Este es generado y configurado manualmente, realizando la alineación del modelo relacional con el modelo ontológico definido en la actividad de modelado. Luego, la generación de tripletas RDF puede brindar dos salidas según se especifique. La primera genera todas las tripletas que conforman el grafo RDF en su totalidad. La segunda contempla la creación de tripletas RDF a partir de los cambios detectados en la BDR. De lo descrito anteriormente, se obtienen las tripletas RDF y se conoce a *priori* el tipo de operación que implicó el cambio en la BDR. Por lo tanto, es posible realizar las consultas SPARQL de inserción, eliminación o modificación correspondientes, que se encargarán de reflejar los cambios identificados en la BDR en el grafo RDF (Reyes-Álvarez et al. 2014).

¹ <http://semanticweb.org/wiki/VirtuosoRdfViews>

1.3.1.4. Morph

Morph² (formalmente llamado ODEMapster) es un motor RDB2RDF desarrollado por el Ontology Engineering Group, que sigue las especificaciones R2RML³. Morph es compatible con la actualización de datos (RDB a instancias RDF) y la traducción de consultas (SPARQL a SQL). Morph emplea diversas técnicas con el fin de generar consultas SQL eficientes. Morph ha sido probado con consultas reales de varios proyectos españoles y ha demostrado ser uno de los trabajos más eficientes de las herramientas del estado del arte disponibles. Por el momento, Morph funciona con MySQL, PostgreSQL y MonetDB (Priyatna, Corcho y Sequeda 2014).

1.3.1.5. Lenguajes de alineación

Investigaciones recientes demuestran que es posible trasladar del modelo de datos relacional al modelo de datos RDF y viceversa. En este sentido el grupo de trabajo RDB2RDF perteneciente al W3C⁴ propone lenguajes para establecer una alineación entre el modelo de datos relacional y las ontologías. Estos lenguajes de alineación demuestran la posibilidad de realizar operaciones comunes en ambos sentidos tanto: relacional-grafo como grafo-relacional.

Direct Mapping

Direct Mapping define una representación de un grafo RDF de los datos de un BDR. Toma como entrada los datos y el esquema de una BDR y hace corresponder las tablas con las clases de la ontología y los atributos de las tablas con propiedades del vocabulario, generando un grafo RDF. Genera una ontología putativa, la cual no puede ser modificada una vez generada. Las relaciones entre las tablas generan propiedades de objeto. La alineación se hace de forma automática por lo que se pierde personalización al realizar la misma (Hert, Reif y Gall 2011).

R2RML

Es el lenguaje de alineación que recomienda la W3C para la alineación de BDR a RDF. El objetivo es definir un lenguaje de alineación independiente de los datos que se procesen, a través de un proceso semiautomático debido a que se pueden modificar algunos parámetros como los nombres en la alineación. Utiliza ontologías de dominio para modelar los datos empleando tablas lógicas, las cuales

² <https://github.com/fpriyatna/morph>

³ <http://www.w3.org/TR/r2rml/>

⁴ <https://www.w3.org/>

pueden ser una tabla de la BDR, una vista o una consulta SQL válida (Reyes-Álvarez et al. 2014; Hert, Reif y Gall 2011; Das, Sundara y Cyganiak 2012).

D2RQ

D2RQ es un lenguaje de alineación y una plataforma para el tratamiento de BDR como grafos RDF virtuales. Su objetivo consiste en exponer BDR en el ámbito de la Web Semántica para proporcionar acceso a través de consultas SPARQL y datos enlazados (Hert, Reif y Gall 2011). La alineación está expresada en RDF y formalmente definida por un RDF Schema. D2RQ genera vistas en modo de solo lectura, por lo que no ha de ser posible realizar modificaciones ni a la BDR ni al grafo RDF (Eisenberg y Kanza 2012).

eD2R

Es una extensión de D2RMAP cuyo objetivo es abarcar situaciones donde en la alineación intervengan bases de datos altamente estructuradas o que no estén en primera forma normal. La alineación está basada en consultas SQL que extraigan información de la BDR y funciones de transformación que puedan ser aplicadas a los valores extraídos (Hert, Reif y Gall 2011).

R2O

Es un lenguaje extensible y completamente declarativo para describir la alineación entre esquemas de BDR y ontologías implementadas en RDFS u OWL. Se asume que la BDR y el modelo ontológico ya existen. Se utiliza en situaciones donde la similaridad entre la ontología y el modelo de BDR es baja. Está concebido para casos de alineaciones complejas donde un modelo es más rico, más genérico/específico o mejor estructurado que el otro. La alineación está basada en una sintaxis basada en XML (Hert, Reif y Gall 2011).

Relational.OWL

Define una ontología OWL Full para describir el esquema y los datos de una BDR. El objetivo de la aplicación de esta alineación es el intercambio de datos punto a punto entre las bases de datos.

Triplify

Es una aproximación para publicar datos enlazados desde BDR. Está basado en la alineación de respuestas HTTP-URI en consultas de BDR y trasladando las relaciones resultantes en sentencias RDF. La principal motivación de Triplify es que la mayoría de la información en la web está almacenada

de forma estructurada en BDRs pero publicada como HTML por aplicaciones web (CMS, Wiki, Blog). La alineación Triplify está implementada como scripts PHP (Hert, Reif y Gall 2011).

R3M

Es el lenguaje de alineación de la plataforma OntoAccess. Soporta el acceso de lectura y escritura. Emplea una sintaxis basada en RDF que contiene la alineación de tablas a clases y de atributos a propiedades (Hert, Reif y Gall 2011).

1.3.1.6. Resumen de las herramientas estudiadas para la transformación.

Para la transformación se estudiaron varias herramientas, entre las que destacan Virtuoso RDF Views, D2RQ++, UpRDF y Morph. De ellas UpRDF es la herramienta más novedosa debido a que permite la transformación de los datos al modelo de datos RDF a partir de BDR de manera incremental. Esta herramienta permite actualizar los cambios realizados en una BDR hacia un grafo RDF sin tener que generar el grafo en su totalidad.

Tabla II: Comparativa de las herramientas estudiadas para la transformación

Características	Virtuoso	UpRDF	D2RQ++	Morph
Lenguaje de implementación	C	Java	Java	Java
Licencia	Código Abierto y Comercial	Código Abierto	Código Abierto	Código Abierto
Fuente de datos	BDR	BDR	BDR	BDR
Lenguaje de alineación	Virtuoso RDF Views	R2RML	D2rq	R2RML
Tipo de transformación	Completa	Incremental	Completa	Completa

1.3.2. Enfoques para el enlazado

En el enlazado se identifican aquellos grafos RDF que contengan recursos iguales o que puedan proveer información extra para otro grafo. De esta manera se da cumplimiento a la necesidad establecida en el cuarto principio de los datos enlazados, el cual se refiere a la creación o inclusión de enlaces entre diversas fuentes de datos. Para el caso de los recursos equivalentes se emplea el enlace owl:sameAs y para adicionar más información se utiliza rdfs:seeAlso.

1.3.2.1. Silk

Si bien existen cada vez más herramientas disponibles para la publicación de datos enlazados en la web, todavía existen escasas herramientas para el establecimiento de enlaces RDF a otras fuentes de datos. Silk⁵ es un marco de trabajo para el descubrimiento de enlaces. Utiliza el lenguaje declarativo Silk - Link Specification Language (Silk-LSL) para la especificación de enlaces. Este lenguaje utiliza expresiones heurísticas para decidir la relación semántica que existe entre dos entidades. Entre las principales características que presenta este marco de trabajo se encuentran (Volz et al. 2009):

- Soporta la generación de enlaces *owl:sameAs* así como otros tipos de enlaces RDF.
- Posee un lenguaje declarativo flexible para la especificación de las condiciones de los enlaces.
- Puede ser empleado en entornos distribuidos sin replicar los conjuntos de datos.
- Puede ser usado en situaciones donde los términos de diferentes vocabularios se encuentren mezclados y no exista consistencia entre los esquemas RDFS y OWL.
- Implementa varios métodos para el almacenamiento en caché, indexación y preselección de entidades para aumentar el rendimiento y disminuir la carga de la red.

1.3.2.2. LIMES

LIMES⁶ (Link Discovery Framework for Metric Spaces) es una aproximación para el descubrimiento de enlaces entre fuentes de datos enlazados ubicados en conjuntos de datos donde las distancias entre todos sus miembros se encuentran definidas. Esta aproximación utiliza la desigualdad triangular para calcular estimaciones pesimistas de la similitud entre las instancias. LIMES puede filtrar un gran número de parejas de instancias que no cumplan las condiciones deseadas por los usuarios. Presenta dos novedosos algoritmos para la aproximación eficiente de distancias. El primer algoritmo se ocupa de crear una cantidad n (donde n es un número natural definido en la configuración del enlazado) de grupos con la mayor disimilitud posible. Luego el segundo algoritmo crea enlaces sobre las instancias de los conjuntos ordenados cuya distancia sea inferior al umbral deseado (Ngomo y Auer 2011).

1.3.2.3. KnoFuss

KnoFuss es un sistema que presenta un aprendizaje no supervisado donde la configuración del enlazado se realiza utilizando algoritmos genéticos. Dichos algoritmos comparan las entidades de dos grafos RDFs utilizando funciones de aptitud y métodos de similitud. Los subconjuntos con los resultados relevantes de ambos grafos son seleccionados utilizando consultas SPARQL. Cada regla de decisión

⁵ <https://www.assembla.com/spaces/silk/wiki/Home>

⁶ <http://aksw.org/Projects/LIMES.html>

de los candidatos es usada como una entrada de la herramienta KnoFuss para crear el correspondiente conjunto de enlaces (Nguyen y Ichise 2015).

Entre las principales tareas que se realizan en la arquitectura KnoFuss mencionadas en (Nikolov et al. 2008) se encuentran:

- *Correferenciación*: La salida de esta tarea es un conjunto de asignaciones entre los individuos que se consideran idénticos.
- *Detección de conflictos*: Este estado identifica nuevos datos cuando la integración de nuevos datos viola las restricciones ontológicas y diagnostica cada inconsistencia. La salida de esta tarea contiene el conjunto de sentencias que describen cada inconsistencia.
- *Resolución de inconsistencias*: Esta tarea integra cada dato en la base de conocimiento objetivo, la cual incluye procesamiento de inconsistencias.

La ventaja de KnoFuss es que este sistema no requiere que los datos se encuentren etiquetados para el entrenamiento. Sin embargo, ya que el valor de aptitud es calculado utilizando un pseudo valor de la exactitud real, la configuración del aprendizaje puede contener alineaciones incorrectas y por lo tanto reducir el rendimiento (Nguyen y Ichise 2015).

1.3.2.4. Resumen de las herramientas estudiadas para el enlazado

Entre las herramientas estudiadas para el enlazado destacan Silk, LIMES y KnoFuss. De estas herramientas Silk ofrece la solución más adecuada para el enlazado de grafos RDF almacenados en un almacén de tripletas RDF. Mientras que Silk implementa varios métodos para aumentar su rendimiento, KnoFuss y LIMES emplean configuraciones para la alineación que pueden contener enlaces incorrectos.

Tabla III: Comparativa de las herramientas estudiadas para el enlazado

Características	Silk	LIMES	KnoFuss
Lenguaje de implementación	Java	Java	Java
Licencia	Código Abierto	Código Abierto	Código Abierto
Entrada	Grafos RDF, archivo xml	Grafos RDF	Grafos RDF
Salida	Archivo ntriples, uri actualizada	Archivo TTL	Archivo TTL
Soporte para owl:sameAs	SI	SI	SI

Lenguaje de alineación	R2RML	-	-
Almacenamiento en caché	SI	NO	NO
Indexación	SI	SI	NO

1.3.3. Enfoques para la publicación

Esta actividad tiene como objetivo hacer accesible en la web los grafos RDF previamente generados y enlazados. Para la publicación de grafos RDF en la web existen tres formas conocidas. La primera es proveer un *SPARQL Endpoint* como método de acceso hacia el grafo RDF publicado en algún almacén de tripletas. La segunda forma es mediante un *Linked Data Frontend*. Este tipo de herramientas proporcionan una vía para la utilización de los datos no solo para los humanos sino también para las computadoras. Finalmente, se pueden publicar los grafos RDF directamente en la web utilizando un archivo con las tripletas serializadas en algún formato conocido para este propósito. El objetivo de esta aproximación es construir servidores con clientes inteligentes, reduciendo la baja disponibilidad de los *SPARQL Endpoints* públicos. Publicar metadatos que describan a los grafos RDF resulta de utilidad tanto para los productores como para los consumidores de datos enlazados (Hidalgo Delgado 2015).

1.3.3.1. D2R Server

D2R Server es una herramienta para la publicación de bases de datos relacionales en la web semántica. Usa el lenguaje de alineación D2RQ para alinear el contenido de la base de datos al modelo RDF. Utiliza los navegadores HTML y RDF para visualizar el contenido de la base de datos. Siguiendo los principios de los datos enlazados D2R Server asigna una URI a cada entidad que se describe en la base de datos. El nombre de dichas URIs es el mismo tanto para la interfaz web como para el grafo RDF. El grafo resultante es accedido a través de SPARQL 1.1 sobre el protocolo SPARQL (Bizer y Cyganiak 2006).

1.3.3.2. Pubby

Pubby⁷ es un *Linked Data Frontend* que permite mostrar los datos en formato HTML o RDF según las características del navegador web a utilizar. Esto hace que la información sea parte de la web, en lugar de permitir el acceso solo a través de SPARQL. Una de las características principales que presenta Pubby es que permite la dereferenciación de URI. La dereferenciación de URI se conoce como la recuperación de una representación de un recurso identificado por una URI. Esto hace posible que

⁷ <http://wifo5-03.informatik.uni-mannheim.de/pubby/>

todos los datos descritos en un recurso puedan ser representados mediante un navegador. Entre las principales características de esta aproximación se encuentran: (Cyganiak y Bizer 2008)

- Proporciona una interfaz HTML simple que muestra los datos disponibles sobre cada recurso.
- Se encarga de manejar las redirecciones 303 y la negociación de contenido.
- Compatible con los servidores de contenidos Tomcat y Jetty.
- Incluye una extensión de metadatos para incluir metadatos a los datos proporcionados.
- Permite la dereferenciación de URIs.

1.3.3.3. WESO DESH

WESO DESH⁸ es un frontend empresarial de datos enlazados para *SPARQL endpoints* escrito en Java. El mismo permite entre otras características: (Sifaqui 2012)

- Negociación de contenido utilizando las redirecciones 303 del protocolo HTTP.
- Almacena los resultados en caché.
- La salida nativa es HTML y RDFa.
- Las URIs se definen a través de expresiones regulares.
- Permite la ejecución de múltiples tipos de consultas SPARQL tales como (CONSTRUCT, ASK and DESCRIBE).
- Administración a través de una interfaz de usuario gráfica.
- Permite complejos patrones de URIs usando expresiones regulares.

1.3.3.4. ELDA

ELDA⁹ es una implementación Java de la especificación Linked Data API (LDA) desarrollada por Epimorphics Ltd. Proporciona una forma configurable para acceder a los datos RDF usando URLs REST simples que son traducidos en consultas hacia un *SPARQL Endpoint*. Esto hace que sea de manera más fácil para los desarrolladores que utilizan tecnologías web tales como JavaScript y JSON acceder a los datos y mostrarlos en un navegador web. En esta aproximación el desarrollador de la API especifica como trasladar las URLs en consultas. Con su implementación vienen algunos ejemplos preconstruidos que le permiten experimentar con el estilo de consulta y empezar la construcción de nuevas especificaciones (Yu y Liu 2015).

⁸ <https://code.google.com/archive/p/weso-desh/>

⁹ <https://code.google.com/archive/p/elda/>

1.3.3.5. Resumen de las herramientas estudiadas para la publicación

Entre las herramientas estudiadas para la publicación de datos enlazados destacan Pubby, D2R Server, Weso Desh y ELDA. Como se muestra en la siguiente tabla Pubby provee la mayoría de las características comunes presentes en las otras herramientas para la publicación de grafos RDF. Pubby también permite dereferenciar las URIs para que los humanos puedan acceder a la descripción de cualquier recurso mediante su URI.

Tabla IV: Comparativa de las herramientas estudiadas para la publicación

Características	Pubby	D2R Server	Weso Desh	ELDA
Lenguaje de implementación	Java	Java	Java	Java
Licencia	Código Abierto	Código Abierto	Código Abierto	Código Abierto
Entrada	Grafo RDF	Grafo RDF	Grafo RDF	Grafo RDF
Provee Linked Data interface	SI	SI	SI	SI
URIs dereferenciables	SI	-	-	-
Interfaz HTML	SI	NO	NO	SI
Redirecciones 303	SI	SI	NO	-
Servidores compatibles	Tomcat, Jetty	Tomcat	Tomcat	Tomcat

1.4. Conclusiones parciales

Durante la revisión de la literatura se estudiaron varios lenguajes de alineación entre los cuales destacan R2RML y *Direct Mapping* por ser los recomendados por el W3C para la alineación entre bases de datos relacionales y grafos RDF. La herramienta UpRDF es la más adecuada para realizar la transformación BDR2RDF debido a que no existe evidencia en la literatura consultada de otra herramienta que permita realizar la actualización de manera incremental.

Durante el enlazado la obtención de un resultado correcto se encuentra estrechamente relacionado con la métrica de comparación definida. El rendimiento de esta actividad depende del número de comparaciones realizadas entre los elementos de los grafos que se desean enlazar. La herramienta Silk implementa mecanismos para mejorar el rendimiento del enlazado y provee un lenguaje descriptivo que permite establecer criterios de comparación según las necesidades de los usuarios.

Las herramientas analizadas en la publicación corresponden a la segunda forma para la publicación de datos enlazados. La herramienta Pubby es el *Linked Data Frontend* que posee la mayoría de las

características comunes de las herramientas para la publicación de datos enlazados y además permite la dereferenciación de las URIs.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

2.1. Introducción

En el presente capítulo se abordan las características del componente referentes al diseño, arquitectura e implementación de la propuesta de solución. Se definen los métodos, herramientas y metodologías empleadas, así como la descripción de los mismos. Para una mejor comprensión se detallan por separado las actividades de transformación, enlazado y publicación.

2.2. Descripción general de la propuesta

Con el desarrollo de un componente que integre las actividades de transformación, enlazado y publicación de grafos RDF, debe mejorar el tiempo en que se realizan estas actividades. El mismo realizará la transformación de datos provenientes de una base de datos relacional hacia el modelo de datos RDF. Permitirá enlazar entidades que compartan características comunes entre dos grafos RDF. También publicará los datos para que puedan ser accedidos desde un *SPARQL Endpoint* o un navegador web.

2.3. Metodología de desarrollo de software

La metodología adoptada por el grupo de investigación de web semántica es una variación para la UCI de la metodología “Proceso Unificado Ágil” (AUP) en su escenario No 4. Este escenario solo puede modelar el sistema con Historias de Usuario (HU). Se decidió optar por esta metodología debido a que el negocio a informatizar se encuentra bien definido y el cliente siempre acompañará al equipo de desarrollo para convenir los detalles de los requisitos, implementarlos, probarlos y validarlos (Navarro y Rosa [2016]).

2.4. Tecnologías empleadas

Las herramientas y tecnologías propuestas por el grupo de investigación de web semántica para la realización del componente son: el marco de trabajo para el desarrollo de aplicaciones web *Grails* v.2.5.3. Como lenguaje de programación *Groovy*¹⁰. Como entorno de desarrollo integrado (IDE) *IntelliJ IDEA*¹¹ v.14.0.1, como sistema gestor de bases de datos (SGBD) *PostgreSQL*¹² v.9.4 y como sistema

¹⁰ <http://www.groovy-lang.org/>

¹¹ <http://www.jetbrains.org>

¹² <http://www.postgresql.org.es/>

gestor de versiones Git. Como lenguaje de consultas de grafos RDF, *SPARQL 1.1 Update* y el *almacén de tripletas RDF Apache Jena Fuseki*.

Las herramientas UpRDF, Silk y Pubby fueron seleccionadas para la realización de las actividades de transformación, enlazado y publicación respectivamente, por ser las más adecuadas para dar solución al problema en cuestión.

2.5. Técnicas de captura de requisitos

Antes que los requisitos puedan ser analizados, modelados o especificados, deben ser recogidos a través de un proceso de captura de requisitos (Pressman Roger 2002). Para la captura de requisitos las técnicas utilizadas fueron: tormenta de ideas y prototipado.

2.6. Arquitectura del componente

A continuación, se presenta la arquitectura general del proyecto Biblioteca Digital Semántica. Este proyecto está compuesto por cuatro procesos de los definidos en las guías metodológicas para la publicación de datos enlazados. La presente investigación se enmarca en el proceso de publicación.

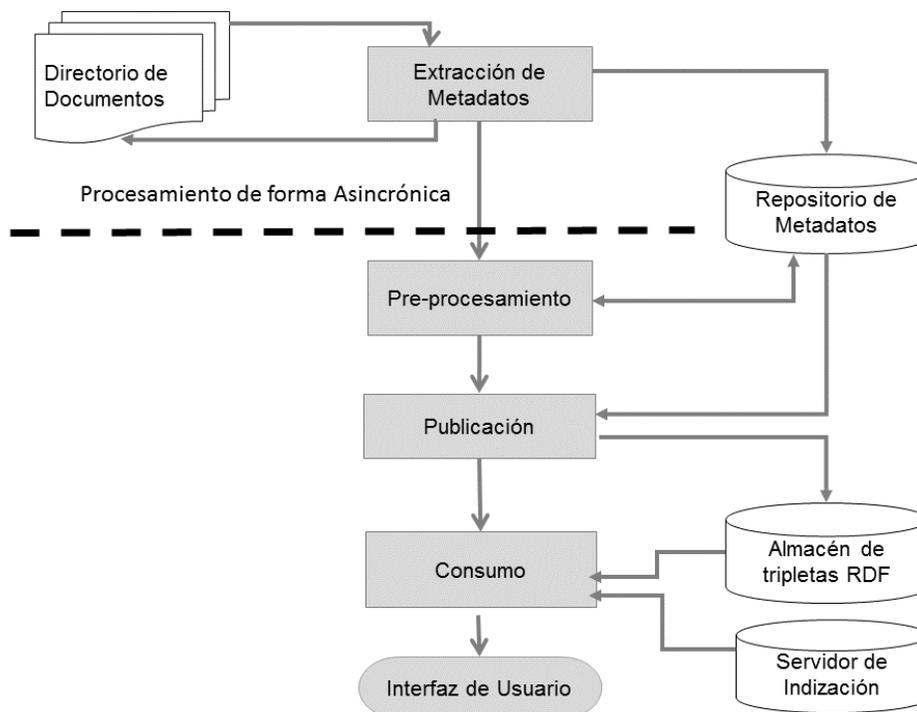


Fig. 3: Arquitectura del proyecto Biblioteca Digital Semántica

El método propuesto consta con tres fases para realizar la publicación de datos enlazados. Este método está basado en el patrón arquitectónico tuberías y filtros donde los componentes llamados filtros se encuentran conectados por tuberías que transmiten datos de un componente al siguiente (Pressman Roger 2002). Las fases propuestas son las siguientes:

1. *Transformación* de los datos almacenados en el modelo relacional al modelo de datos RDF. También se incluye la actualización incremental del grafo RDF a través los cambios ocurridos en la BDR.
2. *Enlazado* de recursos del grafo con otros recursos existentes en grafos RDF similares.
3. La *publicación* del grafo permitirá que cualquier aplicación que pueda procesar el modelo de datos RDF pueda hacer uso de sus datos.

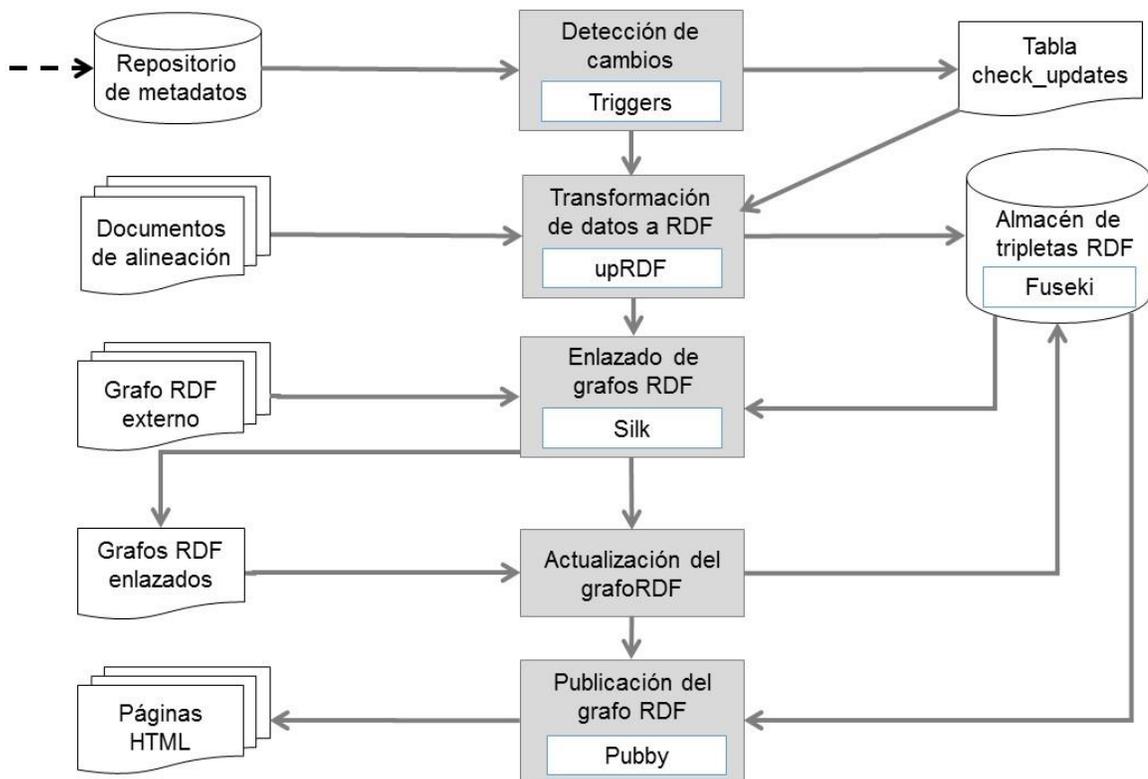


Fig. 4: Arquitectura del componente Publicación de datos enlazados

2.6.1. Transformación

El modelo propuesto tiene como entrada una base de datos relacional que almacena los metadatos de publicaciones científicas. Para la detección de cambios en la BDR se crea una nueva tabla nombrada *check_updates*. Los atributos de esta tabla son: *table_name*, *id_row* y *type_of_change*, los cuales hacen referencia al nombre de la tabla, el identificador de la fila donde se realizó el cambio, así como

el tipo de modificación realizado (inserción, modificación o eliminación) respectivamente. Se crean funciones que inserten los datos de dicha tabla y se crean los *triggers* asociados a cada tabla que invocan a las correspondientes funciones. A cada tabla especificada se asocian dos *triggers*, uno para las operaciones de inserción y modificación y otro para la eliminación. La diferencia entre ambos radica en la captura del identificador de la fila relacionada al cambio.

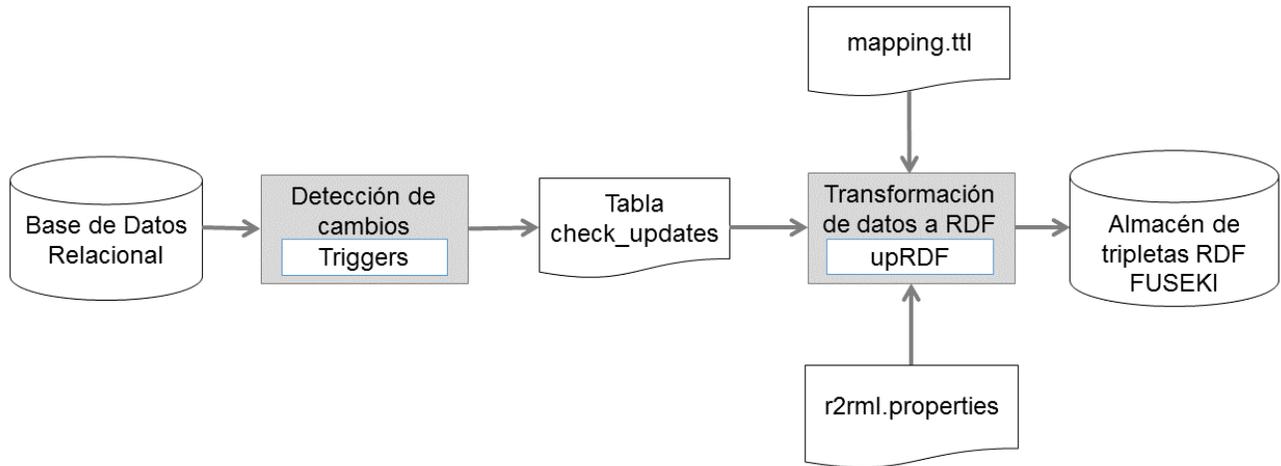


Fig. 5: Método para la transformación de grafos RDF. Fuente: elaboración propia.

Para la generación de las tripletas RDF se necesita el documento de alineación (*mapping.ttl*). Este documento es generado manualmente y está basado en el lenguaje de alineación R2RML (ver figura 6). En este documento se establece la alineación del modelo relacional con el modelo ontológico definido. Se presentarán los parámetros de configuración de este documento a través de una interfaz de usuario que facilita el acceso y modificación de dichos parámetros.

```

1  @prefix map: <http://localhost/metharto/#>
2  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
3  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
4  @prefix rr: <http://www.w3.org/ns/r2rml#>.
5  @prefix vocab: <vocab/>.
6  @prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
7  @prefix foaf: <http://xmlns.com/foaf/0.1/>.
8  @prefix swro: <http://swrc.ontoware.org/ontology#>.
9  @prefix dc: <http://purl.org/dc/elements/1.1/>.
10 @prefix bibo: <http://purl.org/ontology/bibo/>.
11 @prefix fabio: <http://purl.org/spar/fabio/>.
12 @prefix ns: <http://localhost/metharto/#>.
13 @prefix : <http://localhost/metharto/#>.
14
15 map:author
16   rr:logicalTable [ rr:tableName "author" ];
17   rr:subjectMap [ rr:class foaf:Person; rr:template 'http://localhost/metharto/author/{id}' ];
18   rr:predicateObjectMap [
19     rr:predicate foaf:name;
20     rr:objectMap [ rr:column "nom_name" ];
21   ];
22   rr:predicateObjectMap [
23     rr:predicate swro:affiliation;
24     rr:objectMap [ rr:column "affiliation" ];
25   ];
26   rr:predicateObjectMap [
27     rr:predicate rdfs:label;
28     rr:objectMap [ rr:termType rr:literal; rr:template 'author #{id}' ];
29   ];
30   rr:predicateObjectMap [
31     rr:predicate foaf:name;
32     rr:objectMap [ rr:column "name" ];
33   ];
34
35 map:set
36   rr:logicalTable [ rr:tableName "set" ];
37   rr:subjectMap [ rr:class fabio:ItemCollection; rr:template 'http://localhost/metharto/set/{id}' ];
38   rr:predicateObjectMap [
39     rr:predicate vocab:set_spen;
40     rr:objectMap [ rr:column "set_spen" ];
41   ];

```

Fig. 6: Fragmento de archivo de alineación *mapping.ttl*. Fuente: elaboración propia.

Sus valores podrán ser editados a través de un formulario donde se encuentren los detalles de configuración del archivo *mapping.ttl*. Luego, la generación de tripletas RDF puede brindar dos salidas según se especifique. La primera genera todas las tripletas que conforman el grafo RDF en su totalidad. La segunda contempla la creación de tripletas RDF a partir de los cambios detectados en la BDR. Esta opción permitirá indicar si se desea actualizar el grafo o crearlo a partir de los datos de la BDR.

La herramienta UpRDF es la encargada de realizar la transformación del modelo relacional a RDF. Para realizar la alineación es necesario generar un documento empleando el lenguaje R2RML. Este documento referencia a la estructura de las tablas de la BDR para obtener los datos contenidos en ellas. Para la configuración de este documento se generará una interfaz de usuario que contenga sus datos en un formulario que permita a los mismos ser editados.

El modelo ontológico constituye un paso fundamental en la alineación desde la base de datos relacional al modelo de datos RDF. Su objetivo es la creación o reutilización de ontologías necesarias para la modelación de los datos. Esta tarea es esencial, a partir que un correcto modelado de los datos, permitirá posteriormente la generación del grafo RDF. Cualquier detalle que se omita o cualquier ontología necesaria que se deje de plasmar en el modelo ontológico, influirá posteriormente en la publicación de los metadatos como datos enlazados, afectando el proyecto desarrollado (Reyes-Álvarez et al. 2014).

Luego de haber obtenido las tripletas RDF y conociendo a priori el tipo de operación que implicó el cambio en la BDR, UpRDF realiza las consultas SPARQL correspondientes. Estas consultas de inserción, eliminación o modificación se encargarán de reflejar los cambios detectados de la BDR en el grafo RDF. UpRDF no realiza de manera automática un chequeo a la tabla *check_updates*, por lo cual su ejecución estará regida de forma periódica a través de tareas programadas que se ejecutan en intervalos de tiempo de 10 segundos. La salida de esta fase está constituida por un grafo RDF que contenga los mismos datos almacenados en la base de datos relacional.

2.6.2. Enlazado

Para realizar la actividad de enlazado entre el grafo generado en la fase anterior y otros grafos contenidos en el almacén de tripletas RDF se emplea la herramienta Silk. Esta herramienta permite realizar un enlazado de manera automática entre dos entidades. Los parámetros de configuración serán

editados empleando el lenguaje declarativo (Silk-LSL) a través de una interfaz de usuario. Las tripletas generadas por el enlazado se incluirán al primer grafo RDF al concluir el enlazado.

2.6.2.1. Configuración para enlazar

La estructura del archivo de configuración para realizar la actividad de enlazado está compuesta por cuatro bloques obligatorios:

- Definición de prefijos: se definen los prefijos que aparecen en los dos grafos.
- Definición de fuentes: se define las fuentes de datos, así como la dirección física donde se encuentran.
- Especificación de enlaces: establece las métricas de similitud entre dos elementos y los umbrales que deben cumplir las métricas para realizar el enlazado.
- Definición de salida: representa el destino donde los enlaces generados son almacenados.

La solución propuesta para realizar esta actividad consiste en la integración de esta herramienta con el componente que se desea desarrollar. Esta es una aplicación desarrollada en Java y que permite la utilización de sus funcionalidades a través de una librería Java. Para su correcto funcionamiento se debe colocar el archivo de configuración para la alineación en la estructura interna del proyecto.

2.6.3. Publicación

Para publicar los datos almacenados en el almacén de tripletas se empleará la herramienta Pubby. Esta herramienta es un *Linked Data Frontend* para *SPARQL Endpoint*. Recibe como entrada el archivo de configuración *config.ttl* que será editado a través de una interfaz de usuario. El archivo *config.ttl* se encuentra ubicado en la carpeta WEB-INF de la herramienta Pubby. La misma se actualizará de manera automática a medida que existan modificaciones en el grafo RDF.

La estructura del archivo de configuración *config.ttl* es la siguiente:

- Definición de prefijos:

Esta es la primera sección del archivo de configuración y debe contener los prefijos definidos en el modelo ontológico realizado en el proceso de Modelado. Además, debe contener el siguiente prefijo:

@prefix conf: <http://richard.cyganiak.de/2007/pubby/config.rdf#>.

- Configuración de Pubby

La configuración de la herramienta define las siguientes sentencias:

<> a conf:Configuration; Esta sentencia inicia la configuración de la herramienta.

conf:projectName "Biblioteca Digital Semántica"; Esta sentencia permite escribir el nombre que se mostrará en la vista de la herramienta. En este ejemplo el nombre es Biblioteca Digital Semántica.

conf:projectHomepage <localhost:8080/index.html>; Esta sentencia permite establecer un enlace para el nombre del proyecto. Este enlace se define generalmente hacia la página de inicio de la aplicación.

conf:webBase <http://localhost:8080/bds/>; Esta sentencia determina la ruta base donde se encuentra publicada la herramienta en el servidor.

conf:defaultLanguage "en"; Esta sentencia determina el lenguaje por defecto de la herramienta.

conf:indexResource <http://localhost:8080/bds/autor/1>; Esta sentencia indica el recurso que se mostrará por defecto cuando se acceda a la url base donde se encuentra publicada la herramienta.

- **Configuración de los grafos RDF**

conf:dataset [; Esta sentencia inicia la descripción del grafo que se desea publicar.

conf:sparqlEndpoint <http://localhost:3030/bds/sparql>; Esta sentencia indica la url del SPARQL Endpoint donde se encuentra el grafo RDF.

conf:datasetBase <http://localhost:8080/bds/>; Esta sentencia indica donde se publicará el grafo RDF.

conf:webResourcePrefix ""; Esta sentencia indica a partir de que recurso común se publicará el grafo RDF. Por defecto el grafo se publicará en su totalidad.

conf:fixUnescapedCharacters "(),!\$&*+;=@"; Esta sentencia especifica los caracteres no válidos para la herramienta.

];: Se concluye con los delimitadores de cierre.

2.7. Estándares de codificación

Los estándares de codificación son patrones de programación que no están orientados a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Existen diversas formas de estandarizar el código de acuerdo a cada lenguaje, sin embargo, se puede hacer una generalización de los puntos más importantes a considerar, entre los que se encuentran:

- **Comentarios:** Se aconseja el uso de comentarios en línea para facilitar la comprensión del código. Cada método contiene un comentario al inicio con la descripción de la funcionalidad. Si el método es muy largo se incluirán comentarios intermedios que describirán bloques lógicos dentro de la funcionalidad.

- Funcionalidades y atributos: Debe tener un nombre significativo en concordancia con la funcionalidad. Para los identificadores se utilizará la notación CamelCase, comenzando la primera palabra con letra inicial minúscula y el resto con letra inicial Mayúscula.
 - Ejemplo de función: ejecutarSilk()
 - Ejemplo de atributo: rutaAbsoluta
- Estructuras de control: Se utilizarán llaves para mejorar la legibilidad y disminuir la posibilidad de errores lógicos al agregar nuevas líneas de código. Se utilizarán los nombres i, j y k para los contadores de los ciclos.
- Clases: Para los nombres de las clases se empleará la notación UpperCamelCase, comenzando siempre cada palabra con letra inicial mayúscula y las restantes con minúscula.
 - Ejemplo de clase controladora: TransformacionController
 - Ejemplo de clase del dominio: Persona

2.8. Requisitos funcionales de software

Tabla V: Requisitos funcionales

No.	Requisito funcional
1	Detectar cambios en la BDR
2	Transformar grafo RDF
3	Enlazar grafo RDF
4	Editar archivo de alineación
5	Editar archivo de transformación
6	Editar archivo de enlazado
7	Editar archivo de publicación

2.9. Historias de usuario

Tabla VI: Historia de Usuario Detectar cambios en la BDR

Número: 1		Nombre del requisito: Detectar cambios en la BDR.	
Programador: Luis Manuel Pérez Batista		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 7 días	
Riesgo en Desarrollo: Alto		Tiempo Real: 7 días	

Descripción: Se recogen los cambios realizados en la BDR mediante *triggers* y se almacenan en la tabla `check_updates`. Los cambios pueden ser de inserción, modificación o eliminación.

Prototipo de interfaz: No aplica

Tabla VII: Historia de Usuario Transformar grafo RDF

Número: 2	Nombre del requisito: Transformar grafo RDF.
Programador: Luis Manuel Pérez Batista	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 28 días
Riesgo en Desarrollo: Alto	Tiempo Real: 21 días
Descripción: El componente transformará los datos contenidos en la base de datos relacional al modelo de datos RDF y los incluirá en el grafo contenido en el almacén de tripletas Fuseki. Esta actividad se ejecutará en intervalos de diez segundos utilizando tareas programadas.	
Prototipo de interfaz: No aplica	

Tabla VIII: Historia de Usuario Enlazar grafo RDF

Número: 3	Nombre del requisito: Enlazar grafo RDF
Programador: Luis Manuel Pérez Batista	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 50 días
Riesgo en Desarrollo: Alto	Tiempo Real: 42 días
Descripción: El usuario selecciona el botón Enlazar en la vista Enlazado y se crean los enlaces entre dos grafos RDF definidos en la configuración. El resultado será actualizado de manera automática en el grafo especificado como salida.	
Prototipo de interfaz:	

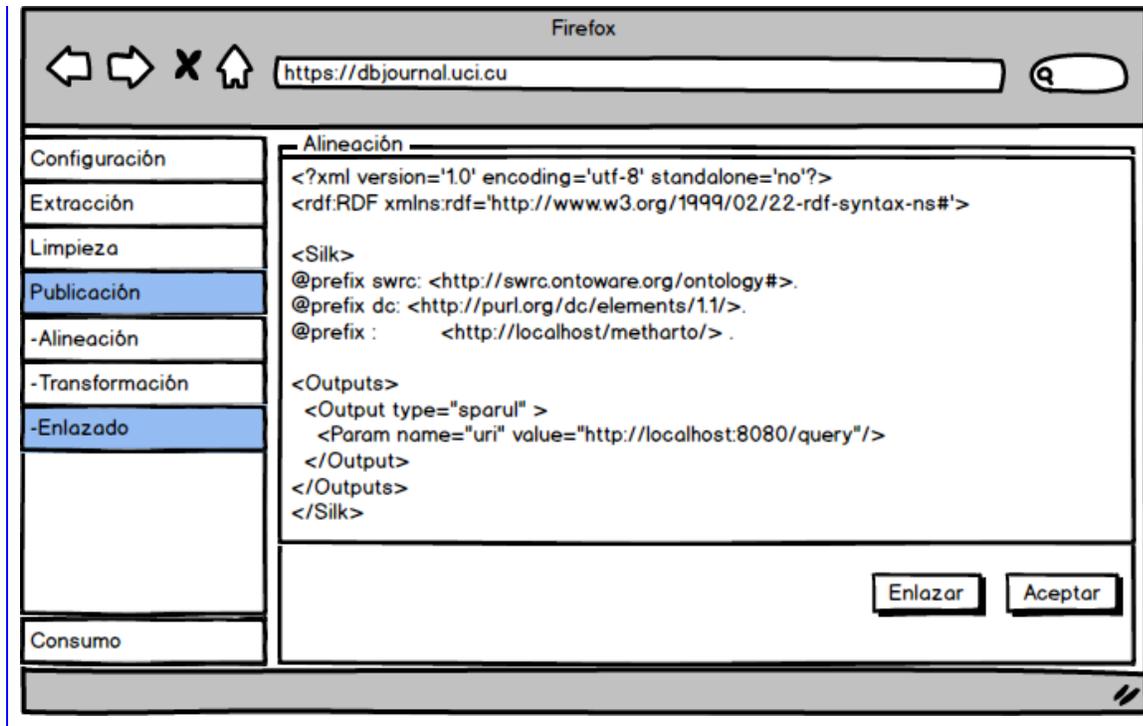


Tabla IX: Historia de Usuario Editar archivo de alineación

Número: 4	Nombre del requisito: Editar archivo de alineación.	
Programador: Luis Manuel Pérez Batista	Iteración Asignada: 1	
Prioridad: Baja	Tiempo Estimado: 14 días	
Riesgo en Desarrollo: Bajo	Tiempo Real: 14 días	
Descripción: El usuario puede configurar los parámetros del archivo de alineación R2RML a través de la vista Alineación. Una vez realizados los cambios el usuario debe proceder a seleccionar el botón “Aceptar” para que estos tengan efecto. Luego, se muestra la vista previa del grafo RDF con datos obtenidos de la base de datos relacional.		
Prototipo de interfaz:		

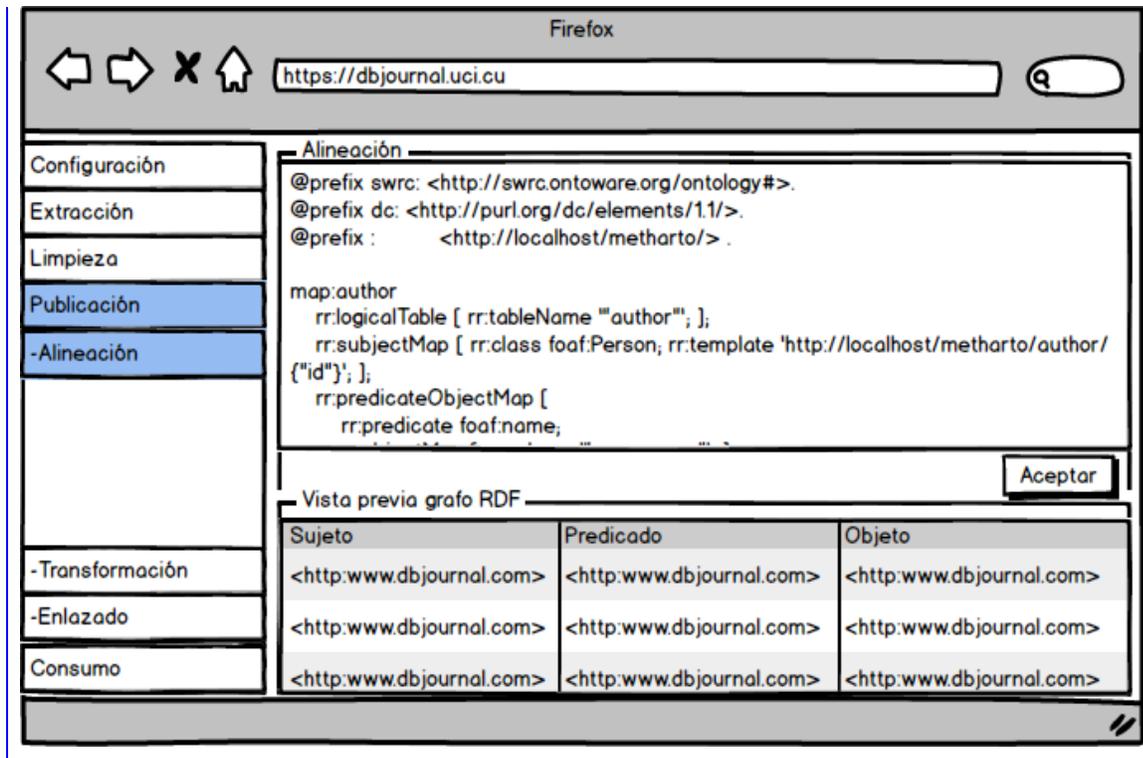


Tabla X:Historia de Usuario Editar archivo de transformación

Número: 5		Nombre del requisito: Editar archivo de transformación	
Programador: Luis Manuel Pérez Batista		Iteración Asignada: 1	
Prioridad: Baja		Tiempo Estimado: 14 días	
Riesgo en Desarrollo: Bajo		Tiempo Real: 14 días	
Descripción: El usuario puede configurar los parámetros de la transformación a través de la vista Transformación. Una vez realizados los cambios el usuario debe proceder a seleccionar el botón “Aceptar” para que estos tengan efecto.			
Prototipo de interfaz:			

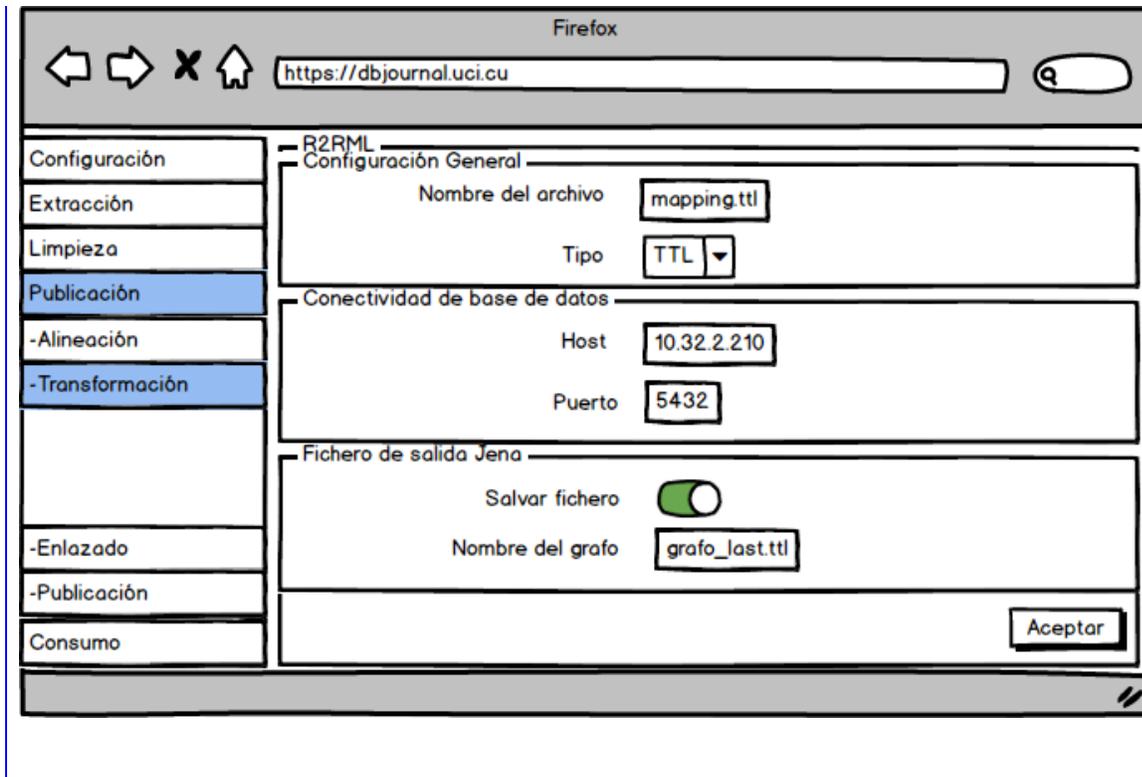


Tabla XI: Historia de Usuario Editar archivo de enlazado

Número: 6	Nombre del requisito: Editar archivo de enlazado.
Programador: Luis Manuel Pérez Batista	Iteración Asignada: 1
Prioridad: Baja	Tiempo Estimado: 14 días
Riesgo en Desarrollo: Bajo	Tiempo Real: 14 días
Descripción: El usuario puede configurar los parámetros del enlazado a través de la vista Enlazado. Una vez realizados los cambios el usuario debe proceder a seleccionar el botón “Aceptar” para que estos tengan efecto.	
Prototipo de interfaz:	

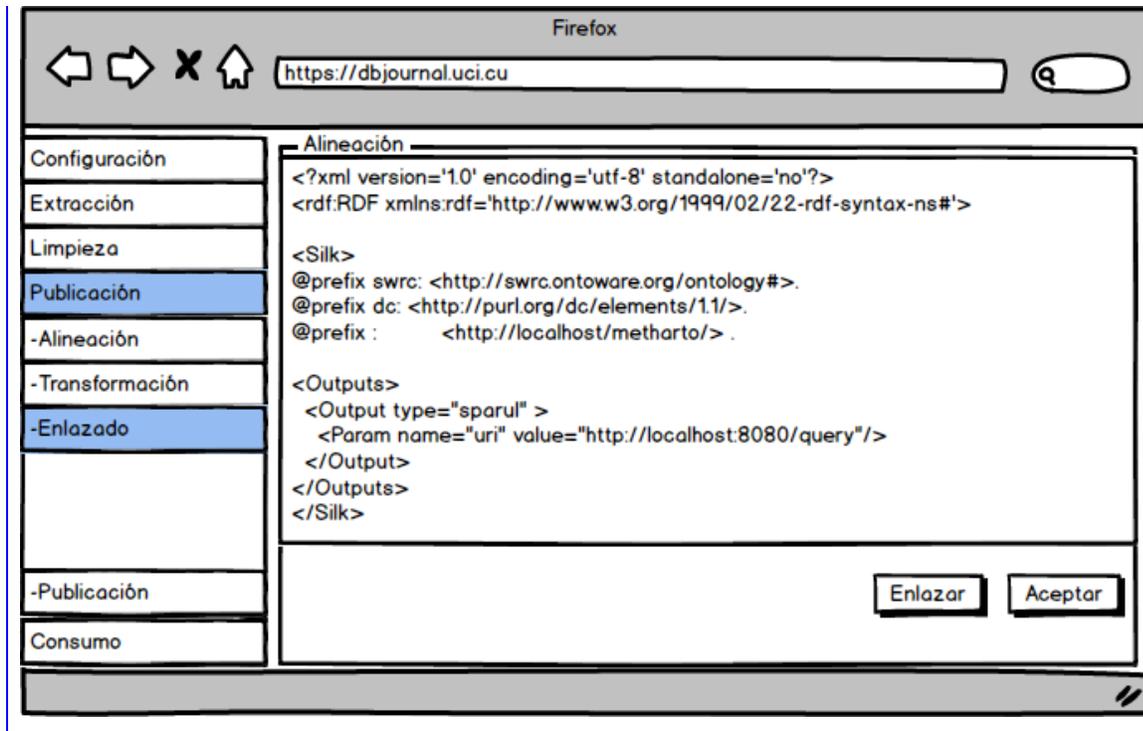
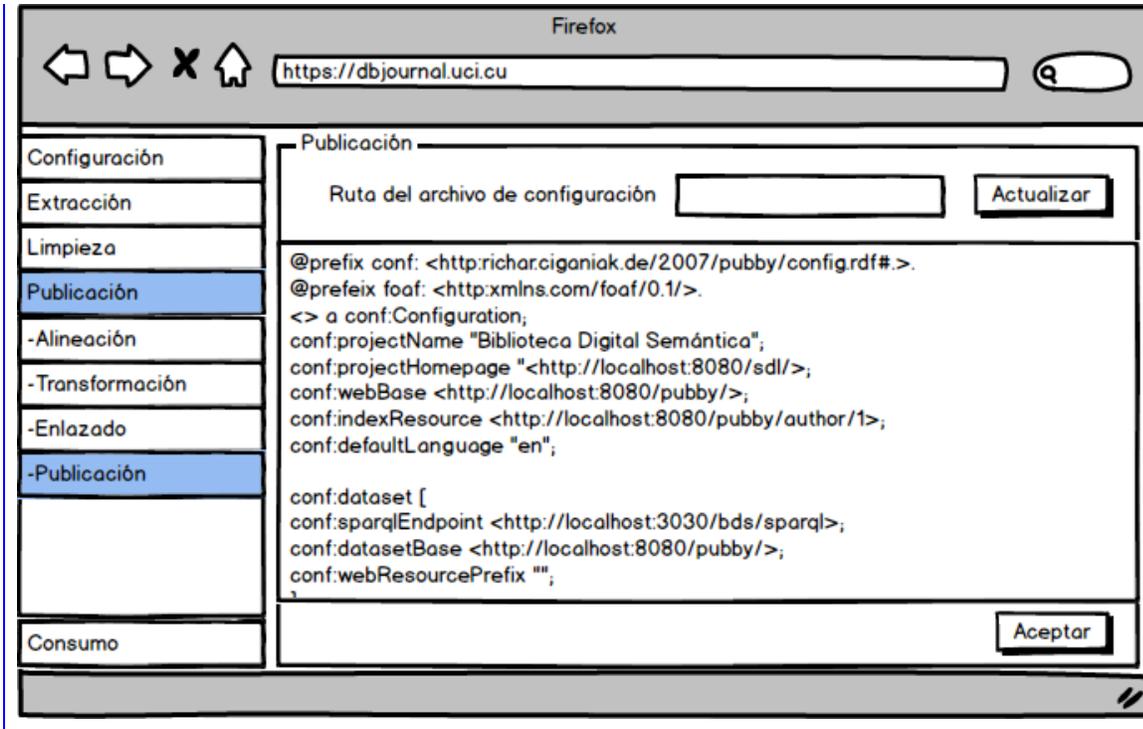


Tabla XII: Historia de Usuario Editar archivo de publicación

Número: 7	Nombre del requisito: Editar archivo de publicación.
Programador: Luis Manuel Pérez Batista	Iteración Asignada: 1
Prioridad: Baja	Tiempo Estimado: 14 días
Riesgo en Desarrollo: Bajo	Tiempo Real: 10 días
Descripción: El usuario puede configurar los parámetros de la publicación a través de la vista Publicación. Una vez realizados los cambios el usuario debe proceder a seleccionar el botón “Aceptar” para que estos tengan efecto.	
Prototipo de interfaz:	

2.10. Requisitos no funcionales

Los requerimientos no funcionales son condiciones que debe cumplir un sistema para satisfacer un contrato o una especificación. Están regidos por las necesidades del usuario para resolver un problema o conseguir un beneficio determinado. Se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida y la representación de datos que se utilizan en las interfaces del sistema. Estos



requerimientos son de gran significación en la aceptación del software, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema (Moreno y Marciszack 2013).

2.10.1. Software

No.	Requisito no funcional de software
1	Navegador web Mozilla Firefox a partir de su versión 44.
2	Máquina virtual de Java v.1.7 o superior.
3	Servidor de aplicación Tomcat Server v.7.
4	Herramienta UpRDF para la alineación, creación y actualización incremental de los grafos RDF.
5	Herramienta Silk para el enlazado de los grafos RDF.
6	Frontend Pubby para la publicación del grafo.

2.10.2. Hardware

Del lado del servidor

No.	Requisito no funcional de hardware
-----	------------------------------------

1	Microprocesador Intel core i5-2380P CPU 3.10Ghz
2	4 GB de memoria RAM.
3	1 Tera Byte (TB) de disco duro.
4	Tarjeta de red: Fast-Ethernet 100MB/s.

2.10.3. Diseño

No.	Requisito no funcional de diseño
1	Se utilizará Metronic como plantilla para el diseño de las interfaces.
2	En cada interfaz de usuario se emplearán los colores blanco, azul, negro y rojo.
3	Los botones tendrán los colores azul, verde y rojo.
4	Las letras serán de tamaño 14px, color blanco para los menús y negro para el texto y el estilo Open sans y Sans-serif.
5	Para los mensajes de operación satisfactoria se utilizará el color verde, para los de advertencia el color amarillo y para los de error el color rojo.
6	Se utilizará un menú de correderas y se resaltará la casilla seleccionada.

2.11. Conclusiones parciales

El diseño de la arquitectura de software, la selección de la metodología, los estándares de codificación y las herramientas empleadas durante la implementación de la aplicación resultaron necesarias para la consolidación del componente desarrollado. Las Historias de Usuario descritas por el cliente permitieron conocer el nivel de detalle para cada requisito funcional. El componente desarrollado cumple con todas las especificaciones de los requisitos identificados y se ajusta a la arquitectura de software propuesta.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

3.1. Introducción

En este capítulo se realiza la validación del componente propuesto en el capítulo anterior. Se desarrollan los tipos y las técnicas de pruebas empleadas para validar la propuesta de solución. Se realizan pruebas de aceptación para cada HU y pruebas internas al código según describe la metodología AUP-UCI. Se desarrolla un caso de estudio empleando datos almacenados en una base de datos relacional que integra los datos extraídos de artículos científicos. Se realiza un preexperimento con el objetivo de evaluar los tiempos de respuesta de la aplicación al realizar las diferentes tareas que inician las actividades de transformación, enlazado y publicación de grafos RDF. Se realiza un análisis de los resultados obtenidos durante las pruebas realizadas y se emiten criterios respecto a la solución desarrollada.

3.2. Planificación de pruebas

La planificación de las pruebas garantiza que el software desarrollado cumpla con todos los requisitos que le han sido asignados antes de su entrega final. Esta actividad permite realizar revisiones y pruebas a lo largo del desarrollo del software para asegurar la calidad del mismo. Los ingenieros de software afrontan la calidad aplicando métodos técnicos, realizando revisiones técnicas formales y llevando a cabo pruebas de software bien planificadas (Pressman Roger 2002).

3.2.1. Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se utilizará la prueba de unidad orientada a **caja blanca**. Este tipo de prueba es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. La técnica de caja blanca a utilizar será el **camino básico**. Al utilizar esta técnica se garantiza que todas las sentencias del bloque escogido se ejecutan al menos una vez (Pressman Roger 2002).

3.2.2. Pruebas de aceptación

Las pruebas de aceptación permiten que el cliente valide todos los requisitos. Estas pruebas son realizadas por el usuario final en lugar del responsable del desarrollo del sistema. Las pruebas de aceptación realizadas serán pruebas de caja negra. Este tipo de pruebas se centran en los requisitos

funcionales y tienen como objetivo encontrar errores de las siguientes categorías (Pressman Roger 2002):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de estructuras de datos o en acceso a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

3.3. Pruebas de software

Una prueba de software es un proceso de ejecución de un programa con la intención de descubrir un error. Para la realización de las pruebas se generan casos de pruebas que tengan una probabilidad alta de mostrar un error no descubierto hasta entonces (Pressman Roger 2002). De las pruebas definidas por la metodología seleccionada se realizarán las pruebas internas y pruebas de aceptación.

3.3.1. Pruebas de caja blanca

Las pruebas de caja blanca se aplicaron haciendo uso de la técnica del camino básico, con el objetivo de evaluar la complejidad lógica de un diseño procedimental y usar esta medida como guía para la definición de un conjunto básico de caminos de ejecución (Pressman Roger 2002). Esta prueba permite garantizar que en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez.

A continuación, se enumeran y analizan las sentencias de código del método ***obtenerDiezPrimerasTripletasRdf()*** contenido en la clase ***TransformacionController***. Este método retorna una lista enlazada con las diez primeras tripletas contenidas en el grafo RDF.

```

LinkedList<LinkedList<String>> obtenerDiezPrimerasTripletasRdf() {
Properties properties = new Properties() //1
String propertiesFilename = rutaAbsoluta()+'/r2rml.properties'
properties.load(new FileInputStream(propertiesFilename))
String endpointQuery = properties.get('sparql.endpoint.query')
LinkedList<LinkedList<String>> triples = new LinkedList<>()
String sujeto = '', predicado = '', objeto = ''
String sparqlQuery = 'SELECT ?s ?p ?o WHERE { ?s ?p ?o } LIMIT 10'
QueryExecution qe =
    QueryExecutionFactory.sparqlService(endpointQuery, sparqlQuery)
com.hp.hpl.jena.query.ResultSet execSelect = qe.execSelect()
while (execSelect.hasNext()) { //2
    QuerySolution qs = execSelect.next()
    sujeto = qs.getResource('s').getURI()
    predicado = qs.getResource('p').getURI()
    if (qs.get('objeto').isLiteral()) //3
        objeto = qs.getLiteral('o').getString() //4
    else{
        objeto = qs.getResource('o').getURI() //5
    }
    LinkedList<String> triple = new LinkedList<>()
    triple.clear()
    triple.add(sujeto)
    triple.add(predicado)
    triple.add(objeto)
    triples.add(triple) //6
}
return triples //7
}

```

Fig. 7: Función que obtiene las diez primeras tripletas del grafo RDF.

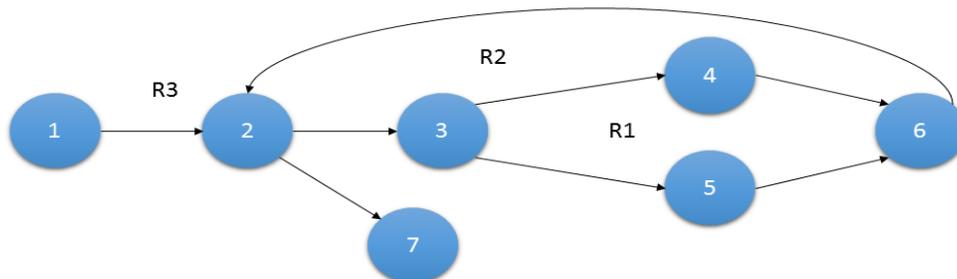


Fig. 8: Grafo de flujo asociado al método obtenerDiezPrimerasTripletasRdf().

Luego se representa el grafo de flujo asociado al procedimiento anterior a través de nodos, aristas y regiones, ver Fig.7.

Cálculo de la complejidad ciclomática del grafo de flujo anterior

$$V(g) = (a - n) + 2$$

$$V(g) = p + 1$$

$$V(g) = r$$

Siendo “V(g)” el valor de la complejidad ciclomática, “a” la cantidad total de aristas, “n” la cantidad total de nodos, “p” la cantidad total de nodos predicados (nodos de los cuales parten dos o más aristas) y “r” la cantidad total de regiones.

$$V(g) = (8 - 7) + 2 = 3$$

$$V(g) = 2 + 1 = 3$$

$$V(g) = 3$$

Al evaluar las fórmulas anteriores se obtiene una complejidad ciclomática igual a 3. Este valor representa el número mínimo de casos de prueba para el procedimiento tratado. Seguidamente se muestran los caminos básicos que puede tomar el procedimiento durante su ejecución.

Camino básico #1: 1-2-3-4-6-2-7

Camino básico #2: 1-2-3-5-6-2-7

Camino básico #3: 1-2-7

Se procede a ejecutar los casos de prueba para cada camino básico determinado en el grafo de flujo.

Tabla XIII: Caso de prueba para el camino básico #1

CASO DE PRUEBA CAMINO BÁSICO #1	
Descripción:	Este procedimiento ejecuta una consulta en el lenguaje SPARQL. El grafo RDF será el especificado en el archivo de transformación mediante la URL del endpoint_query.
Condición de ejecución:	Debe existir al menos una tripleta RDF donde el objeto sea un “Literal”
Entrada:	Sin parámetros
Resultados esperados:	Una lista con un máximo de diez tripletas RDF correspondientes al grafo especificado en el archivo de transformación.
Resultado:	Satisfactorio

Tabla XIV: Caso de prueba para el camino básico #2

CASO DE PRUEBA CAMINO BÁSICO #2

Descripción:	Este procedimiento ejecuta una consulta en el lenguaje SPARQL. El grafo RDF será el especificado en el archivo de transformación mediante la URL del endpoint_query.
Condición de ejecución:	Debe existir al menos una tripleta RDF donde el objeto sea un "Recurso"
Entrada:	Sin parámetros
Resultados esperados:	Una lista con un máximo de diez tripletas RDF correspondientes al grafo especificado en el archivo de transformación.
Resultado:	Satisfactorio

Tabla XV: Caso de prueba para el camino básico #3

CASO DE PRUEBA CAMINO BÁSICO #3	
Descripción:	Este procedimiento ejecuta una consulta en el lenguaje SPARQL. El grafo RDF será el especificado en el archivo de transformación mediante la URL del endpoint_query.
Condición de ejecución:	No debe existir ninguna tripleta RDF en el grafo
Entrada:	Sin parámetros
Resultados esperados:	Un lista vacía
Resultado:	Satisfactorio

3.3.2. Resultados de las pruebas unitarias

Las pruebas unitarias se realizaron empleando el marco de trabajo Groovy JUnit. Este marco de trabajo permite realizar pruebas internas al código fuente escrito en el lenguaje de programación Groovy. Las pruebas unitarias se realizaron a las 13 funcionalidades de mayor complejidad, de las cuales solo 3 resultaron tener resultados no satisfactorios en la primera iteración. Los resultados satisfactorios representan un 95% del total de pruebas unitarias realizadas.



Fig. 9: Resultados de las pruebas unitarias en la primera iteración

Todas las no conformidades detectadas fueron corregidas de manera satisfactoria. Los resultados de la segunda iteración se muestran en la figura 9.

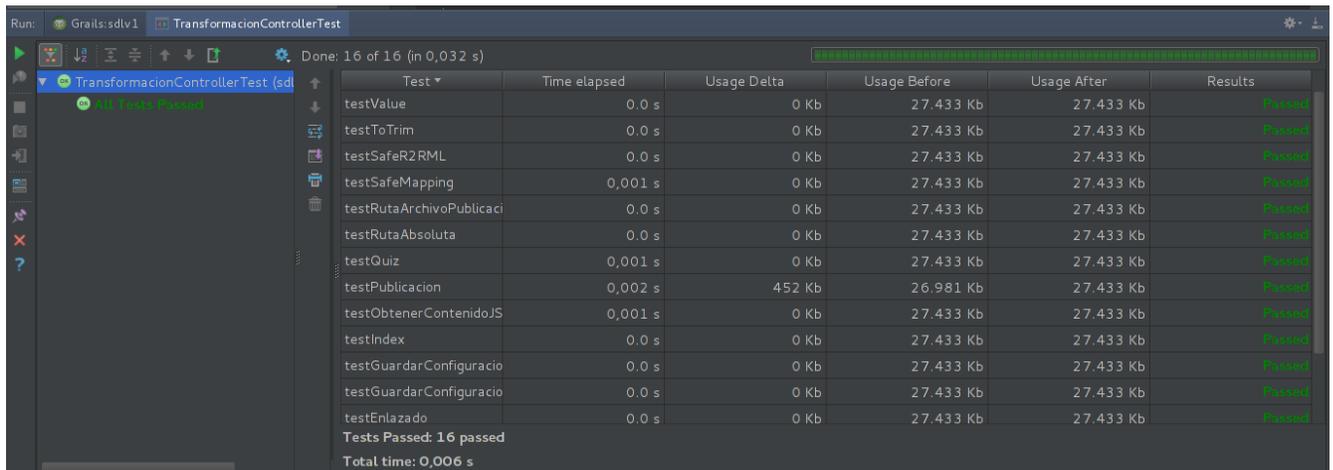


Fig. 10: Resultados de las pruebas unitarias en la segunda iteración.

3.3.3. Pruebas de caja negra

Para realizar las pruebas de caja negra se empleó el método de partición equivalente. Este método divide el campo de entrada en clases de datos de los que se derivan casos de prueba (Pressman Roger 2002). En este trabajo todas las HU son sometidas a pruebas de aceptación.

Tabla XVI: Caso de prueba de aceptación CP-01

Código: CP-01	Historia de Usuario: 1
Nombre: Caso de prueba detectar cambios en la BDR	

<p><i>Descripción:</i> En este caso de prueba se verifica el procedimiento que se lleva a cabo cuando ocurre alguna modificación en la BDR. La descripción de los cambios son almacenados en la tabla <i>check_updates</i>.</p>		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>
Insertar un país	1, Cuba	<ol style="list-style-type: none"> 1. Se registra el cambio en la tabla <i>check_updates</i>. 2. Se insertan las tripletas correspondientes en el grafo RDF.
Actualizar un país	Venezuela	<ol style="list-style-type: none"> 1. Se registra el cambio en la tabla <i>check_updates</i>. 2. Se actualizan las tripletas correspondientes en el grafo RDF.
Eliminar un país	1	<ol style="list-style-type: none"> 1. Se registra el cambio en la tabla <i>check_updates</i>. 2. Se eliminan las tripletas correspondientes en el grafo RDF.
Insertar un documento	1, título, resumen, 2016-06-01, 1, 1, 1, url	<ol style="list-style-type: none"> 1. No se debe de registrar el cambio en la tabla <i>check_updates</i>, debido a que el caso particular de los documentos, solo se guardarán cuando su estado sea publicado.
Modificar un documento	1, título, resumen, 2016-06-01, 1, 1, 3, url	<ol style="list-style-type: none"> 1. Se registra el cambio en la tabla <i>check_updates</i> y en el caso particular de los documentos el tipo de cambio es de inserción porque el documento acaba de ser publicado.(El id 3 corresponde con el estado publicado) 2. Se insertan las tripletas correspondientes en el grafo RDF.
Eliminar un documento	1	<ol style="list-style-type: none"> 1. Se registra el cambio en la tabla <i>check_updates</i>. 2. Se eliminan las tripletas correspondientes en el grafo RDF.
<p><i>Evaluación de la prueba:</i> Satisfactoria</p>		

Tabla XVII: Caso de prueba de aceptación CPA-02

Código: CP-02	Historia de Usuario: 2	
Nombre: Caso de prueba Transformar grafo RDF		
Descripción: Se verifica el procedimiento que se realiza cuando se ejecuta la herramienta UpRDF. Los datos contenidos en la tabla <i>check_updates</i> deben ser transformados al modelo RDF y almacenados en el grafo RDF contenido en el servidor Fuseki.		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>
Transformar un formato	formato, 1, Insert	<ol style="list-style-type: none"> 1. La herramienta UpRDF verifica el cambio en la tabla <i>check_updates</i>. 2. Se insertan las tripletas correspondientes en el grafo RDF, teniendo en cuenta todos los campos de la tupla con el id igual a 1 en la tabla <i>formato</i>.
Transformar un documento	documento,1, Update	<ol style="list-style-type: none"> 1. La herramienta UpRDF verifica el cambio en la tabla <i>check_updates</i>. 2. Se modifican las tripletas correspondientes en el grafo RDF, teniendo en cuenta todos los campos de la tupla con el id igual a 1 en la tabla <i>documento</i>.
Transformar una referencia	referencia,1, Delete	<ol style="list-style-type: none"> 1. La herramienta UpRDF verifica el cambio en la tabla <i>check_updates</i>. 2. Se eliminan las tripletas correspondientes en el grafo RDF, teniendo en cuenta todos los campos de la tupla con el id igual a 1 en la tabla <i>referencia</i>.
Evaluación de la prueba: Satisfactoria		

Tabla XVIII: Caso de prueba de aceptación CPA-03

Código: CP-03	Historia de Usuario: 3

<i>Nombre:</i> Caso de prueba Enlazar grafo RDF		
<i>Descripción:</i> Se verifica el procedimiento que se realiza cuando se ejecuta la herramienta Silk. Según las especificaciones del archivo de enlazado se crean enlaces entre los elementos equivalentes entre el grafo correspondiente a la biblioteca digital semántica y otros grafos RDF.		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>
Seleccionar el botón <i>enlazar</i> de la vista enlazado	No aplica	La herramienta Silk compara las etiquetas de los grafos a enlazar. Se crean tripletas donde el sujeto y el predicado son los elementos que se enlazan y el predicado es owl:sameAs.
<i>Evaluación de la prueba:</i> Satisfactoria		

Tabla XIX: Caso de prueba de aceptación CPA-04

Código: CP-04		
Historia de Usuario: 4		
<i>Nombre:</i> Caso de prueba Configurar archivo de alineación		
<i>Descripción:</i> Se verifica el procedimiento que realiza el usuario modifica los datos del formulario que contiene los datos del archivo de alineación.		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>
Insertar un prefijo en el archivo de alineación	@prefix foaf: <http://xmlns.com/foaf/0.1/>.	1. Se debe insertar el prefijo en el componente de texto que muestra el archivo de alineación.
Seleccionar el botón <i>aceptar</i> de la vista alineación	No aplica	1. Los datos almacenados en el formulario reemplazan los datos del archivo de alineación. 2. Se muestran los datos contenidos en el archivo de alineación. 3. Se muestran un mensaje al usuario donde se indica que el archivo ha sido modificado satisfactoriamente.
<i>Evaluación de la prueba:</i> Satisfactoria		

Tabla XX: Caso de prueba de aceptación CPA-05

Código: CP-05	Historia de Usuario: 5	
<i>Nombre:</i> Caso de prueba Configurar archivo de transformación		
<i>Descripción:</i> Se verifica el procedimiento que realiza el usuario modifica los datos del formulario que contiene los datos del archivo de alineación.		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>
Insertar el nombre del archivo de alineación	mapping.ttl	El nombre se guarda correctamente.
Insertar el nombre del archivo de alineación	mapping	Se debe mostrar un mensaje de error indicando que el nombre del archivo no es válido.
Insertar el nombre del archivo de alineación	Campo vacío	Se debe mostrar un mensaje de error indicando que el campo no debe ser vacío.
Insertar el nombre del espacio de trabajo	http://localhost:8080/bds/	El nombre se guarda correctamente.
Insertar el nombre del espacio de trabajo	Localhost:8080/bds	Se debe mostrar un mensaje de error indicando que el nombre del espacio de trabajo no es una url válida.
Insertar el nombre del espacio de trabajo	Campo vacío	Se debe mostrar un mensaje de error indicando que el campo no debe ser vacío.
Insertar el sparqlEndpointData	http://localhost:3030/bds/data	La url se guarda correctamente.
Insertar el sparqlEndpointData	localhost:3030/bds/data	Se debe mostrar un mensaje de error indicando que el nombre del espacio de trabajo no es una url válida.
Insertar el sparqlEndpointData	Campo vacío	Se debe mostrar un mensaje de error indicando que el campo no debe ser vacío.
Insertar el sparqlEndpointUpdate	http://localhost:3030/bds/update	La url se guarda correctamente.
Insertar el sparqlEndpointUpdate	localhost:3030/bds/update	Se debe mostrar un mensaje de error indicando que el nombre del espacio de trabajo no es una url válida.

Insertar el sparqlEndpointUpdate	Campo vacío	Se debe mostrar un mensaje de error indicando que el campo no debe ser vacío.
Insertar el sparqlEndpointQuery	http://localhost:3030/bds/query	La url se guarda correctamente.
Insertar el sparqlEndpointQuery	localhost:3030/bds/query	Se debe mostrar un mensaje de error indicando que el nombre del espacio de trabajo no es una url válida.
Insertar el sparqlEndpointQuery	Campo vacío	Se debe mostrar un mensaje de error indicando que el campo no debe ser vacío.
<i>Evaluación de la prueba:</i> Satisfactoria		

Tabla XXI: Caso de prueba de aceptación CPA-06

Código: CP-06	Historia de Usuario: 6	
<i>Nombre:</i> Caso de prueba Configurar archivo de enlazado		
<i>Descripción:</i> Se verifica el procedimiento que realiza el usuario modifica los datos del formulario que contiene los datos del archivo de enlazado.		
<i>Acción a probar:</i>	<i>Datos de entrada:</i>	<i>Resultados esperados:</i>
Insertar un umbral para la comparación	1	Se debe insertar el valor del umbral en el componente de texto que muestra el archivo de enlazado.
Seleccionar el botón <i>aceptar</i> de la vista enlazado	No aplica	<ol style="list-style-type: none"> 1. Los datos almacenados en el formulario reemplazan los datos del archivo de alineación. 2. Se muestran los datos contenidos en el archivo de alineación. 3. Se muestran un mensaje al usuario donde se indica que el archivo ha sido modificado satisfactoriamente.
<i>Evaluación de la prueba:</i> Satisfactoria		

Tabla XXII: Caso de prueba de aceptación CPA-07

Código: CP-07		
Código: CP-07	Historia de Usuario: 7	
Nombre: Caso de prueba Configurar archivo de publicación		
Descripción: Se verifica el procedimiento que realiza el usuario modifica los datos del formulario que contiene los datos del archivo de enlazado.		
Acción a probar:	Datos de entrada:	Resultados esperados:
Insertar la url del archivo de publicación	localhost/files/config.ttl	La url se almacena correctamente.
Insertar la url del archivo de publicación	config	Se muestra un mensaje de error indicando que la url no es válida.
Editar el archivo de publicación	conf:datasetBase <http://localhost:8080/pubby>;	Se debe editar el valor en el componente de texto que muestra el archivo de publicación.
Seleccionar el botón <i>aceptar</i> de la vista enlazado	No aplica	<ol style="list-style-type: none"> 1. Los datos almacenados en el formulario reemplazan los datos del archivo de alineación. 2. Se muestran los datos contenidos en el archivo de alineación. 3. Se muestran un mensaje al usuario donde se indica que el archivo ha sido modificado satisfactoriamente.
Evaluación de la prueba: Satisfactoria		

3.3.4. Resultados

Se realizaron un total de 40 casos de prueba de aceptación (caja negra), de ellos 10 resultaron no satisfactorios, lo cual representa el 25% del total de casos de pruebas de caja negra realizados, ver Fig. 10. Las no conformidades detectadas por los casos de prueba fueron mitigadas luego de dos iteraciones de prueba.



Fig. 11: Resultados de las pruebas de aceptación.

3.4. Caso de estudio

Con el objetivo de validar la solución propuesta al problema de la investigación se lleva a cabo un diseño experimental. El diseño experimental es una técnica estadística que permite identificar y cuantificar las causas de un efecto dentro de un estudio experimental (Dzul Escamilla 2013). El tipo de experimento a utilizar es el preexperimento, debido a que se necesita comparar el estado actual de cómo se realizan las actividades de transformación, enlazado y publicación con el estado de las actividades luego de aplicada la propuesta de solución.

3.4.1. Descripción

Para realizar esta prueba se utilizará la BDR *biblioteca*, la cual contiene una estructura equivalente al grafo que se desea crear. Se realizarán una serie de modificaciones en la BDR y se observará el tiempo empleado por las actividades de transformación, enlazado y publicación, antes y después de aplicado el estímulo. El estímulo empleado es la utilización del componente desarrollado para llevar a cabo estas actividades. Las tareas propuestas para realizar esta prueba son:

- Tarea 1: Insertar un autor
- Tarea 2: Modificar un autor
- Tarea 3: Eliminar un autor
- Tarea 4: Insertar 10 revistas
- Tarea 5: Modificar 10 revistas
- Tarea 6: Eliminar 10 revistas

Tabla XXIII: Propuesta para el diseño experimental.

Fuente de datos	Tareas	Observación inicial	Estímulo	Observación final	% de diferencia
B	T ₁	Ol ₁	E	OF ₁	D ₁
	T ₂	Ol ₂		OF ₂	D ₂
	T ₃	Ol ₃		OF ₃	D ₃
	T ₄	Ol ₄		OF ₄	D ₄
	T ₅	Ol ₅		OF ₅	D ₅
	T ₆	Ol ₆		OF ₆	D ₆

A continuación, se muestra la simbología utilizada en la tabla anterior:

- **B:** Base de datos relacional
- **T_i:** Representa la tarea a realizar donde i identifica el número de la tarea.
- **E:** Utilización del componente desarrollado para la transformación, enlazado y publicación de grafos RDF (TEP-RDF).
- **Ol_i:** Tiempo en segundos que demoran en ejecutarse las actividades de transformación, enlazado y publicación de manera manual.
- **OF_i:** Tiempo en segundos que demoran las actividades de transformación, enlazado y publicación luego de aplicado el estímulo.
- **D_i:** Diferencia en por ciento que representa la observación final respecto a la observación inicial.

$$D_i = \frac{oi - of}{oi} * 100$$

3.5. Análisis de los resultados

Tabla XXIV: Análisis de los resultados.

Fuente de datos	Tareas	Observación inicial	Estímulo	Observación Final	% de diferencia
Base de Datos Relacional	T ₁ : Insertar un autor	18	TEP-RDF	5	72%
	T ₂ : Modificar un autor	15		6	60%
	T ₃ : Eliminar un autor	21		6	71%
	T ₄ Insertar 10 revistas	22		8	64%
	T ₅ Modificar 10 revistas	19		8	58%
	T ₆ Eliminar 10 revistas	20		9	55%

El método propuesto permitió una reducción en el tiempo empleado para realizar las actividades de transformación, enlazado y publicación de grafos RDF. Esta reducción del tiempo tributa a una mayor satisfacción del cliente con el uso del software desarrollado. Para la transformación se logró actualizar el grafo de manera automática utilizando tareas programadas. El enlazado se ejecuta al seleccionar un botón que inicia esta actividad y actualiza el grafo RDF de manera automática. Se logró publicar el

grafo RDF mediante un *SPARQL Endpoint* para que pueda ser utilizado por otras aplicaciones y servicios, así como la publicación del mismo mediante un navegador web para ser consultado por humanos. Esto valida la propuesta de solución de forma satisfactoria.

3.6. Conclusiones

Las pruebas realizadas permitieron detectar las no conformidades presentes durante la implementación del componente. Las deficiencias detectadas fueron mitigadas satisfactoriamente luego de dos iteraciones de pruebas. El diseño de un caso de estudio permitió realizar un preexperimento con el objetivo de validar la propuesta de solución presentada en el capítulo anterior. El análisis del tiempo demostró que el componente desarrollado contribuye a la reducción del tiempo empleado en la realización de las actividades de transformación, enlazado y publicación de grafos RDF.

CONCLUSIONES GENERALES

La revisión de la literatura evidenció que las aproximaciones existentes para la transformación, enlazado y publicación de grafos RDF se realizan de manera independiente. Para la actividad de transformación se necesitan las ontologías definidas durante el modelado de los datos y el lenguaje de alineación que se utiliza. El enlazado pretende incorporar enlaces entre todos los datos aislados en la web y lograr que todos los recursos puedan ser accedidos desde cualquier grafo. Las actividades de publicación y consumo se encuentran estrechamente relacionadas y están enmarcadas en proveer métodos para la recuperación de la información a partir de grafos RDF.

Se propone un componente basado en tres fases para la transformación, enlazado y publicación de grafos RDF. Este método implementa un prototipo funcional siguiendo las restricciones y características definidas haciendo uso de una arquitectura basada en tuberías y filtros.

La propuesta de solución disminuye el tiempo en que se realizan las actividades de transformación, enlazado y publicación de grafos RDF. Lo anterior fue corroborado con la realización de un diseño experimental en el cual se analizaron los tiempos en que se realizaron estas actividades utilizando el componente implementado.

El tiempo que demoran las actividades de transformación, enlazado y publicación dependen del número de pasos para realizar estas actividades y del conjunto de datos que se analicen. La observación realizada demuestra que el componente desarrollado disminuye el tiempo en que se realizan estas actividades en más del 50%.

RECOMENDACIONES

El componente desarrollado utiliza la herramienta Pubby para realizar la publicación de los grafos RDF. Esta herramienta contiene su propia interfaz para mostrar los datos, por lo cual se recomienda desarrollar una interfaz visual que permita mostrar los datos publicados con la herramienta Pubby con un mejor diseño.

REFERENCIAS BIBLIOGRÁFICAS

- ADIDA, B., BIRBECK, M., MCCARRON, S. y PEMBERTON, S., 2008. RDFa in XHTML: Syntax and processing a collection of attributes and processing rules for extending XHTML to support RDF, W3C Recommendation. S.l.: s.n.
- ALLEMANG, D. y HENDLER, J., 2011. Semantic web for the working ontologist: effective modeling in RDFS and OWL [en línea]. S.l.: Elsevier. [Consulta: 21 abril 2016]. Disponible en: https://www.google.com/books?hl=es&lr=&id=_qGKPOIB1DgC&oi=fnd&pg=PP1&dq=semantic+web+for+the+working&ots=-ZkdI-AlmS&sig=7JlbbXzRw6iSLsXaPLSPWoYFE34.
- BARZDINS, J. y KIRIKOVA, M., 2011. Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB & IS 2010 [en línea]. S.l.: IOS Press. [Consulta: 21 abril 2016]. Disponible en: https://www.google.com/books?hl=es&lr=&id=7fYS29re860C&oi=fnd&pg=PR1&dq=Databases+and+Information+Systems+VI:+Selected+Papers+from+the+Ninth+International+Baltic+Conference,&ots=6_VRRjkjUK&sig=yxHx8WCFGdifNSV7MeLrJDM4vJU.
- BECHHOFER, S., 2009. OWL: Web ontology language. Encyclopedia of Database Systems [en línea]. S.l.: Springer, pp. 2008–2009. [Consulta: 2 junio 2016]. Disponible en: http://link.springer.com/10.1007/978-0-387-39940-9_1073.
- BERNERS-LEE, T., 2006. Linked data-design issues. [en línea], [Consulta: 28 abril 2016]. Disponible en: <http://www.citeulike.org/group/8357/article/3421195>.
- BERNERS-LEE, T., HENDLER, J., LASSILA, O. y OTHERS, 2001. The semantic web. Scientific american, vol. 284, no. 5, pp. 28–37.
- BERNERS-LEE, T., MASINTER, L., MCCAILL, M. y OTHERS, 1994. Uniform resource locators (URL). [en línea], [Consulta: 2 junio 2016]. Disponible en: <http://www.hjp.at/doc/rfc/rfc1738.html>.
- BIZER, C. y CYGANIAK, R., 2006. D2r server-publishing relational databases on the semantic web. Poster at the 5th International Semantic Web Conference. S.l.: s.n., pp. 294–309.
- CABRERA FACUNDO, A.M. y COUTÍN DOMÍNGUEZ, A., 2005. Las bibliotecas digitales: Parte I. Consideraciones teóricas. Acimed, vol. 13, no. 2, pp. 1–1.
- CRIADO FERNÁNDEZ, L., 2009. Procedimiento semi-automático para transformar la web en web semántica. [en línea], [Consulta: 21 abril 2016]. Disponible en: <http://e-spacio.uned.es/fez/eserv/tesisuned:IngInf-Lcriado/Documento.pdf>.
- CYGANIAK, R. y BIZER, C., 2008. Pubby-a linked data frontend for sparql endpoints. S.l.: s.n.
- DAS, S., SUNDARA, S. y CYGANIAK, R., 2012. $\{\$R2RML: RDB\ to\ RDF\ Mapping\ Language\}\$$. [en línea], [Consulta: 2 junio 2016]. Disponible en: <http://www.citeulike.org/group/14833/article/11522782>.
- DUESA, A.S., CENTELLES, M., FRANGANILLO, J. y GARCÍA, J.G., 2016. Aplicación del modelo de datos RDF en las colecciones digitales de bibliotecas, archivos y museos de España. Revista española de documentación científica, vol. 39, no. 1, pp. 7.

- DZUL ESCAMILLA, M., 2013. Diseños de la investigación. [en línea], [Consulta: 31 mayo 2016]. Disponible en: <http://repository.uaeh.edu.mx/bitstream/handle/123456789/14915>.
- EISENBERG, V. y KANZA, Y., 2012. D2RQ/update: updating relational data via virtual RDF. Proceedings of the 21st international conference companion on World Wide Web [en línea]. S.l.: ACM, pp. 497–498. [Consulta: 2 junio 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=2188095>.
- GARCÍA, C.H.C. y DELGADO, Y.H., 2015. Herramienta para el consumo de metadatos bibliográficos publicados como datos enlazados. Serie Científica [en línea], vol. 8, no. 1. [Consulta: 21 abril 2016]. Disponible en: <http://publicaciones.uci.cu/index.php/SC/article/view/1656>.
- GARROTE HERNÁNDEZ, A., 2014. APIs semánticas para la web orientada a datos enlazados. [en línea], [Consulta: 21 abril 2016]. Disponible en: <http://gredos.usal.es/jspui/handle/10366/124158>.
- GIANNACCINI, C. y MÜLLER, S., 2015. Burckhardtsource.Org: A Semantic Digital Library. Proceedings of the Third AIUCD Annual Conference on Humanities and Their Methods in the Digital Ecosystem [en línea]. New York, NY, USA: ACM, pp. 6:1–6:7. [Consulta: 3 mayo 2016]. ISBN 978-1-4503-3295-8. DOI 10.1145/2802612.2802631. Disponible en: <http://doi.acm.org/10.1145/2802612.2802631>.
- GÓMEZ-PÉREZ, A., VILA-SUERO, D., MONTIEL-PONSODA, E., GRACIA, J. y AGUADO-DE-CEA, G., 2013. Guidelines for multilingual linked data. Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics [en línea]. S.l.: ACM, pp. 3. [Consulta: 21 abril 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=2479867>.
- GUTIÉRREZ, C., 2007. URI y URL.[En línea- Xml]. Santiago,
- HERT, M., REIF, G. y GALL, H.C., 2011. A comparison of RDB-to-RDF mapping languages. Proceedings of the 7th International Conference on Semantic Systems [en línea]. S.l.: ACM, pp. 25–32. [Consulta: 21 abril 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=2063522>.
- HIDALGO DELGADO, Y., 2015. Marco de trabajo basado en los datos enlazados para la interoperabilidad semántica en el protocolo OAI-PMH. [en línea], [Consulta: 21 abril 2016]. Disponible en: <http://eprints.rclis.org/28755/>.
- HIDALGO DELGADO, Y. y RODRÍGUEZ PUENTE, R., 2013. La web semántica: una breve revisión. Revista Cubana de Ciencias Informáticas, vol. 7, no. 1, pp. 76–85.
- HOGAN, A., UMBRICH, J., HARTH, A., CYGANIAK, R., POLLERES, A. y DECKER, S., 2012. An empirical survey of linked data conformance. Web Semantics: Science, Services and Agents on the World Wide Web, vol. 14, pp. 14–44.
- MARSHALL, C.C., 1997. Annotation: from paper books to the digital library. Proceedings of the second ACM international conference on Digital libraries [en línea]. S.l.: ACM, pp. 131–140. [Consulta: 2 junio 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=263806>.
- MORALES, C.R., SOTO, S.V. y MARTÍNEZ, C.H., 2005. Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web. Actas del III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA2005, pp. 49–56.

- MORENO, J.C. y MARCISZACK, M.M., 2013. La Usabilidad Desde La Perspectiva De La Validación de Requerimientos No Funcionales Para Aplicaciones Web. En: 00002, Argentina, ISSN, pp. 2346–9927.
- MURUGESAN, S., 2007. Understanding Web 2.0. *IT professional*, vol. 9, no. 4, pp. 34–41.
- NAVARRO, F.V. y ROSA, D.P., [sin fecha]. SISTEMA DE SOPORTE A LAS DECISIONES CLÍNICAS RELACIONADAS CON EL DIAGNÓSTICO PRECOZ DE ENFERMEDADES. [en línea], [Consulta: 2 junio 2016]. Disponible en: <http://www.informaticahabana.cu/sites/default/files/ponencias/SLD88.pdf>.
- NGOMO, A.-C.N. y AUER, S., 2011. Limes-a time-efficient approach for large-scale link discovery on the web of data. *integration*, vol. 15, pp. 3.
- NGUYEN, K. y ICHISE, R., 2015. Heuristic-Based Configuration Learning for Linked Data Instance Matching. En: G. QI, K. KOZAKI, J.Z. PAN y S. YU (eds.), *Semantic Technology* [en línea]. S.l.: Springer International Publishing, *Lecture Notes in Computer Science*, 9544, pp. 56-72. [Consulta: 23 abril 2016]. ISBN 978-3-319-31675-8. Disponible en: http://link.springer.com/chapter/10.1007/978-3-319-31676-5_4.
- NIKOLOV, A., UREN, V., MOTTA, E. y DE ROECK, A., 2008. Integration of semantically annotated data by the KnoFuss architecture. *Knowledge Engineering: Practice and Patterns* [en línea]. S.l.: Springer, pp. 265–274. [Consulta: 22 abril 2016]. Disponible en: http://link.springer.com/chapter/10.1007/978-3-540-87696-0_24.
- PRESSMAN ROGER, S., 2002. *Ingeniería del software un enfoque práctico*. S.l.: Mc GrawHill.
- PRIYATNA, F., CORCHO, O. y SEQUEDA, J., 2014. Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph. *Proceedings of the 23rd international conference on World wide web* [en línea]. S.l.: ACM, pp. 479–490. [Consulta: 14 junio 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=2567981>.
- PRUD'HOMMEAUX, E., SEABORNE, A. y OTHERS, 2008. SPARQL query language for RDF. *W3C recommendation*, vol. 15.
- REYES-ÁLVAREZ, L., HIDALGO-DELGADO, Y., MARTÍNEZ-ROJAS, K., DEL MAR ROLDÁN-GARCÍA, M. y ALDANA-MONTES, J.F., 2014. Actualización incremental de grafos RDF a partir de bases de datos relacionales. [en línea], [Consulta: 21 abril 2016]. Disponible en: https://www.researchgate.net/profile/Yusniel_Hidalgo_Delgado/publication/266676265_Actualizacin_incremental_de_grafos_RDF_a_partir_de_bases_de_datos_relacionales/links/543760dd0cf2643ab9889d95.pdf.
- SARACEVIC, T., 2000. Digital library evaluation: Toward evolution of concepts. *Library trends*, vol. 49, no. 2, pp. 350–369.
- SIFAQUI, C., 2012. Power to the people! Introducing open linked data services to the National Legal Database of the Library of the National Congress of Chile. *World Library and Information Congress* [en línea]. S.l.: s.n., [Consulta: 14 junio 2016]. Disponible en: <http://conference.ifla.org/sites/default/files/files/papers/wlic2012/214-sifaqui-en.pdf>.
- SPARQL Query Language for RDF. [en línea], [sin fecha]. [Consulta: 2 junio 2016]. Disponible en: <https://www.w3.org/TR/rdf-sparql-query/>.

- VILA-SUERO, D., VILLAZÓN-TERRAZAS, B. y GÓMEZ-PÉREZ, A., 2013. datos. bne. es: A library linked dataset. *Semantic Web*, vol. 4, no. 3, pp. 307–313.
- VOLZ, J., BIZER, C., GAEDKE, M. y KOBILAROV, G., 2009. Silk-A Link Discovery Framework for the Web of Data. LDOW [en línea], vol. 538. [Consulta: 21 abril 2016]. Disponible en: <http://vsr-mobile.informatik.tu-chemnitz.de/svnproxy/download/publications/doc/2009/06.pdf>.
- WOOD, D., ZAIDMAN, M., RUTH, L. y HAUSENBLAS, M., 2014. *Linked Data* [en línea]. S.l.: Manning Publications Co. [Consulta: 2 junio 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=2616311>.
- YU, L. y LIU, Y., 2015. Using Linked Data in a heterogeneous Sensor Web: challenges, experiments and lessons learned. *International Journal of Digital Earth*, vol. 8, no. 1, pp. 17–37.

ANEXOS

Anexo 1: Interfaces del componente implementado.

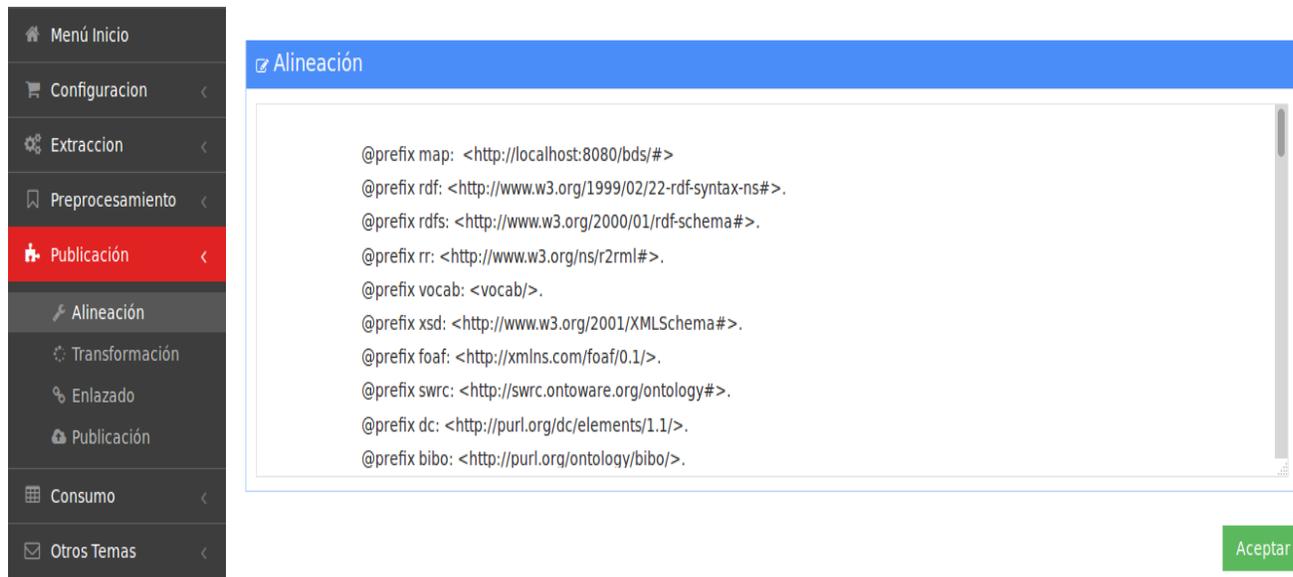


Fig. 12: Interfaz gráfica de la vista Alineación.

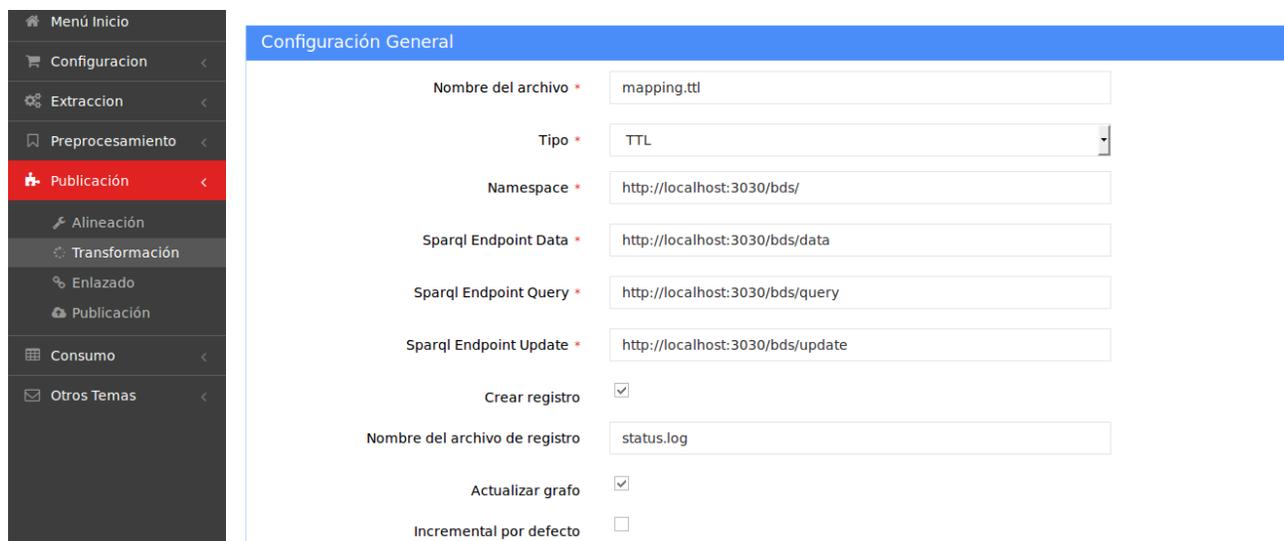


Fig. 13: Interfaz gráfica de la vista Transformación.

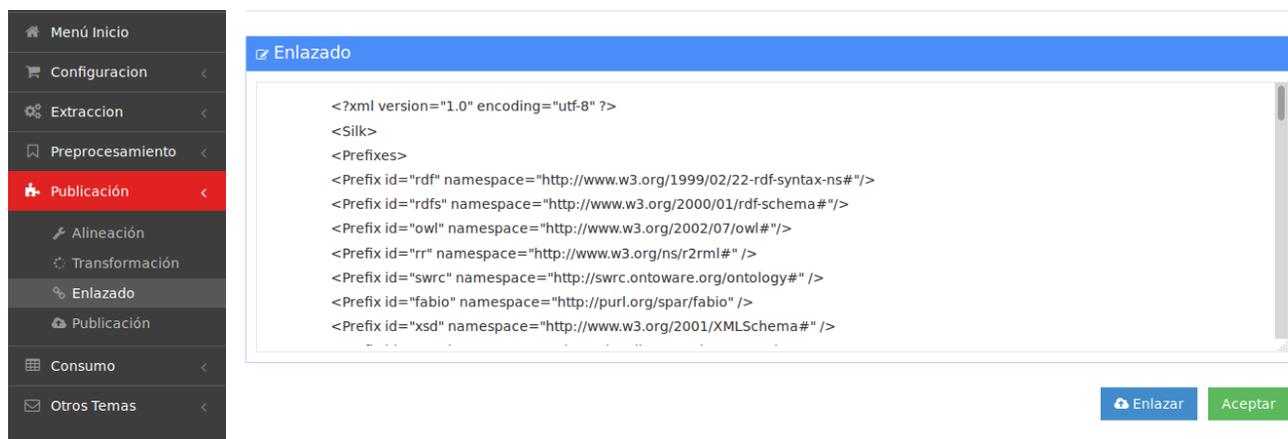


Fig. 14: Interfaz gráfica de la vista Enlazado.

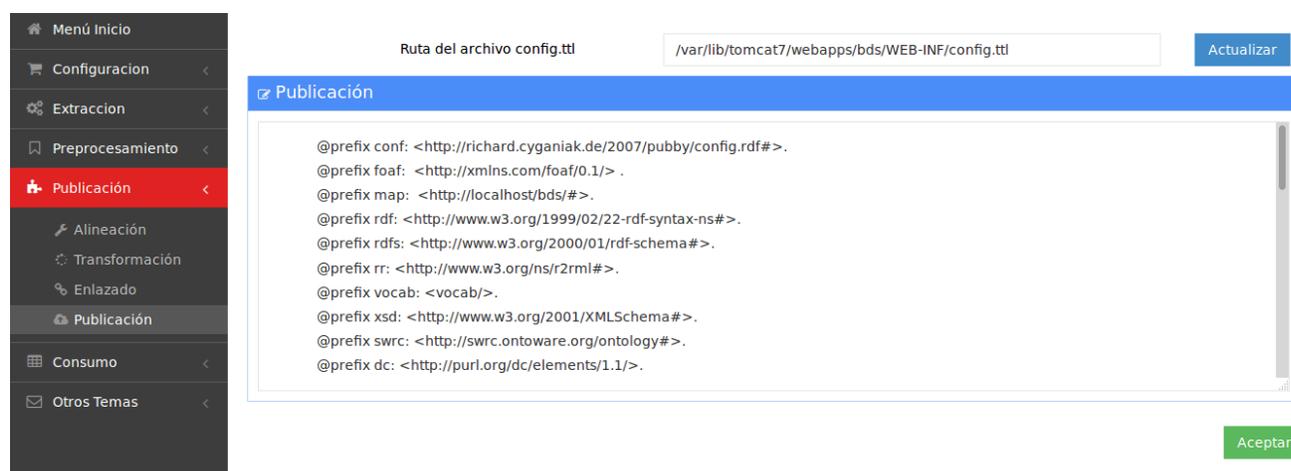


Fig. 15: Interfaz gráfica de la vista Publicación.