Potenciando el consumo de metadatos bibliográficos publicados como datos enlazados

Carlos Heriberto Cordoví García[1], Yusniel Hidalgo Delgado[2]

[1]Departamento de Ciencias Básicas, Facultad 3, Universidad de las Ciencias Informáticas, La Habana, Cuba

[2]Departamento de Programación, Facultad 3, Universidad de las Ciencias Informáticas, La Habana, Cuba

RESUMEN

El éxito de la iniciativa de los datos abiertos enlazados ha incrementado el volumen de información disponible en la Web. Sin embargo, el contenido Web publicado siguiendo esta iniciativa, no puede ser consumido por los usuarios que desconocen las tecnologías de la Web semántica (RDF, SPARQL, Ontologías, etc.), debido a que se requiere la comprensión de la estructura, naturaleza y forma de consultar los datos, lo cual resulta complejo para los usuarios no técnicos. En este trabajo se describe el desarrollo de una aplicación Web que utiliza componentes de la arquitectura de la información y paradigmas de no técnicos. Estos datos se encuentran disponibles a través de un Endpoint (interfaz de acceso a una base de datos de tripletas RDF) el cual puede ser consultado vía HTTP búsqueda, aplicados a la tarea de consumir datos enlazados por usuarios desde un navegador Web. La propuesta de solución en desarrollo permite la búsqueda a texto completo sobre un conjunto de datos bibliográficos y la búsqueda facetada basada en los conceptos más representativos del mismo (facetas). Con la utilización de estos paradigmas de búsqueda se posibilita un consumo de datos enlazados ameno e intuitivo, lo que reduce la barrera técnica que limita a los usuarios en este proceso.

Palabras claves: datos enlazados, Ontología, RDF, SPARQL, Web Semántica

Upgrading bibliographic metadata consumption published as linked data

ABSTRACT

The success of the linked open data initiative has increased the amount of information available on the Web. However, the Web content published under this initiative cannot be consumed by users who are unfamiliar with Semantic Web technologies (RDF, SPARQL, Ontologies, etc.), because they need to understand the structure, provenance and the way in which data are queried, and this can be complex for non-tech-users. In this paper the process development of a Web application is described, which uses components borrowed from Information Architecture and search paradigms applied to the task of consuming Linked Data by non-tech-users. These data are available via a SPARQL endpoint (it is a SPARQL protocol service that enables users to query a knowledge base known as RDF triples database or triplestore), which can be queried through a HTTP protocol from a Web browser. This proposal allows full-text search over a bibliographic dataset and faceted search, based on representative concepts of it (facets). Using these paradigms allows us to consume Linked Data in an intuitive and friendly way, which reduces the technical barrier that limits users in this process.

Keywords: Linked Data, Ontology, RDF, SPARQL, Semantic Web

1. INTRODUCCIÓN

La Web actual ha representado desde su surgimiento un gran avance tecnológico para la humanidad. Ésta ha revolucionado radicalmente el modo en el que de manera tradicional se venían desarrollando las actividades socioeconómicas. Sin embargo, la Web actual posee una serie de limitaciones que impiden aprovechar todas sus potencialidades (Rafael Rodríguez Fuentes, 2013).

Las principales limitaciones de la Web actual (web 2.0) están referidas al formato, la integración y la recuperación de los recursos disponibles en la misma, lo que trae como consecuencia que no existan mecanismos que permitan el procesamiento automático de la información. Esto se debe a que la mayoría de los recursos disponibles en la Web están estructurados en base al formato de hipertexto conocido como HTML, los documentos estructurados con este formato (páginas Web), pueden ser comprendidos fácilmente por los humanos y los navegadores Web, sin embargo, no pueden ser procesados por una máquina y por tanto resulta imposible extraer su valor semántico automáticamente.

Por otro lado, la información en la Web se encuentra dispersa y no existe relación explícita entre los diferentes recursos, esto provoca ambigüedad en la información (2011a). Los problemas de formato e integración provocan que la recuperación de información se vea afectada, lo que se evidencia en los motores de búsquedas más utilizados en la actualidad, los cuales resultan imprecisos y, en muchos casos, no satisfacen las necesidades de búsqueda de los usuarios al responder consultas basadas en palabras claves, no siendo capaces de recuperar la información a partir de consultas expresadas en lenguaje natural (2012).

Ante estas limitaciones y la necesidad de solucionarlas para lograr una evolución de la Web, que permita enfrentar los desafíos de la actual sociedad de la información, Tim Berners-Lee enuncia el concepto de Web Semántica: “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” (Timothy Berners-Lee, 2001).

El tránsito hacia la Web Semántica requiere de una adecuada estructuración e integración de la información, esto propició que en el año 2006 Tim Berners-Lee

enunciara el concepto 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” (2011b).

Los datos enlazados se han convertido en un área de investigación activa en los últimos años. Esto se debe a la necesidad de publicar y consumir datos estructurados en la Web, potenciando de esta manera el desarrollo de la Web Semántica. En la Universidad de las Ciencias Informáticas se está desarrollando el proyecto DBJournal, el mismo consiste en la publicación de metadatos bibliográficos siguiendo los principios de los datos enlazados.

Actualmente los metadatos bibliográficos están publicados como datos enlazados y son accesibles a través de un Endpoint SPARQL, sin embargo, para los usuarios que no conocen sobre las tecnologías de la Web Semántica (RDF, lenguaje de consulta SPARQL, ontologías, vocabularios y metadatos) comúnmente llamados usuarios no técnicos (del inglés lay-user) (2011a) resulta poco intuitiva la utilización de estos datos ya que no conocen la naturaleza y estructura de los mismos, ni la manera de consultarlos.

Cuando un usuario realiza la solicitud de un recurso a través del identificador uniforme de recurso (URI) se retorna información útil relacionada con el recurso. Sin embargo lo común es que la respuesta sea retornada en algún tipo de serialización del formato RDF [RDF/XML, N3, Turtle], entonces el cómo interpretar y utilizar este formato está restringido solamente a usuarios técnicos (del inglés *tech-savvy user*) (2011a) y en ciertos casos, a quienes tienen conocimiento de las tecnologías de la Web Semántica.

En este trabajo se propone una aplicación Web que reutiliza y adapta componentes de la Arquitectura de la Información (AI), para posibilitar que los usuarios que no poseen conocimientos técnicos sobre las tecnologías de la Web Semántica, pero sí de las tecnologías de la informática y las comunicaciones puedan consumir (navegar, consultar, etc.) a través de la búsqueda textual y facetada los metadatos bibliográficos que han sido publicados por el proyecto *DBJournal*. Estos componentes de la arquitectura de la información están presentes en la mayoría de las páginas Web. Ellos son: las facetas de navegación, las “migajas de pan” (del inglés *breadcrumbs*), los menús de navegación, etc.

El resto de este trabajo se ha organizado de la siguiente manera: En la sección 1.1 se aborda sobre los trabajos relacionados con el consumo de datos enlazados. Luego, los materiales y métodos utilizados durante el desarrollo de la propuesta de solución son detallados en la sección 2. La sección 3 presenta las características de la herramienta Web en desarrollo, mientras que la sección 3.1 aborda el uso de los componentes de la AI. La sección 3.2 introduce los paradigmas de búsqueda y las subsecciones 3.2.1 y 3.2.2 abordan los paradigmas de búsqueda textual y búsqueda facetada respectivamente. Por último en la sección 4 se presentan las conclusiones y las proyecciones para el trabajo futuro.

1.1 TRABAJOS RELACIONADOS

La forma de potenciar el consumo (visualización, presentación, utilización) de las fuentes de datos publicadas como datos enlazados de manera intuitiva y amigable para los usuarios de la Web, ha sido una temática abordada por varios proyectos. Las aplicaciones de software desarrolladas con este propósito pueden ser clasificadas básicamente en dos grandes grupos: aplicaciones genéricas y aplicaciones para un dominio específico (2011b). El primer tipo de aplicaciones permite realizar el consumo de datos enlazados desde cualquier dominio temático, de modo que es irrelevante la naturaleza del dato, pueden ser datos geográficos, de ciencias de la vida, bibliotecas, etc.

Existen dos tipos de aplicaciones que pertenecen a la clasificación de aplicaciones genéricas: navegadores de datos enlazados y los motores de búsqueda de datos enlazados. Al primer grupo pertenecen aplicaciones como Disco Hyperdata-Browser (2008b), Tabulator, Marbles1, entre otros. Los navegadores de datos enlazados son útiles pues proveen una experiencia de navegación fluida y similar a la navegación sobre hipertexto de la Web tradicional, sin embargo estos no permiten una visión general del conjunto de datos que se está navegando, ya que solo muestra la información del recurso actual, es decir solo visualiza información del recurso al cual se haya accedido a través de la URI que lo representa.

Los motores de búsquedas de datos enlazados por su parte, mejoran la experiencia del usuario en relación a las tradicionales formas de búsqueda, al realizar la misma sobre datos estructurados. Sin embargo, en estas aplicaciones tampoco es posible

conocer a priori a través de una vista general del conjunto de datos, sus principales recursos, propiedades y valores. A este grupo pertenecen aplicaciones como: Sig.ma (Michele Catasta 2010), VisiNav (Andreas Harth 2010) y Swoogle (Tom Heath 2011).

Las aplicaciones para un dominio específico cubren las necesidades de determinadas comunidades de usuarios, dentro de esta categoría se encuentran los integradores de datos enlazados y otras aplicaciones de dominio específico que incluyen funcionalidades de búsqueda y navegación. Los integradores de datos enlazados son aplicaciones creadas con la finalidad de integrar información desde fuentes heterogéneas (conjuntos de datos entrelazados en la Web de los Datos) para utilizarlas con el propósito de satisfacer las necesidades de información de una determinada comunidad de usuarios. A esta categoría pertenecen aplicaciones como: E.U: US Global Foreign Aid Mashup2 y Paggr3 (2009). Los integradores de datos enlazados son herramientas útiles, puesto que permiten recuperar información interrelacionada entre diversos conjuntos de datos. Sin embargo, estas aplicaciones tampoco resuelven el problema de caracterizar el conjunto de datos a través de una vista general de sus recursos y propiedades.

De modo que las herramientas existentes para el consumo de datos enlazados dificultan que los usuarios no técnicos puedan explorar el conjunto de datos con la finalidad de conocer qué tipos de recursos se encuentran en el mismo, cuáles son sus propiedades y cómo estas se interrelacionan.

2. METODOS

Para el desarrollo de la propuesta de solución se han utilizado tecnologías de la Web Semántica tales como: RDF, SPARQL y Ontologías. Por otro lado se ha hecho uso del lenguaje de programación Web PHP, de la biblioteca de algoritmos EasyRDF4, del marco de trabajo para el desarrollo de aplicaciones JavaScript Jquery y del servidor de propósito general Virtuoso.

RDF (del Inglés Resource Description Framework) es un modelo de datos flexible basado en grafos, constituido como estándar por la W3C, es útil para

describir datos estructurados y sus interrelaciones en un formato procesable por las computadoras. Utiliza ontologías para la descripción formal de los datos en términos de clases y propiedades. Se basa en tripletas de la forma sujeto-predicado-objeto (2008b).

SPARQL es un lenguaje para la realización de consultas sobre uno o múltiples grafos RDF, definido como estándar por la W3C (2008a). Su sintaxis es similar a la del lenguaje SQL aunque orientado a tripletas RDF. Los resultados de las consultas SPARQL pueden ser conjuntos de tripletas RDF, grafos RDF, URIs de recursos o simplemente valores (cadenas de texto, números, etc.).

Una ontología es una especificación explícita y formal sobre 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 (Thomas R. Gruber, 1993).

EasyRDF es una biblioteca de algoritmos escrita en PHP, que facilita la producción y consumo de datos almacenados en formato RDF. La carga e inserción de datos sobre un almacén de grafos RDF se realiza a través de la clase EasyRDF_GraphStore, que implementa funcionalidades para gestionar una colección de grafos RDF vía HTTP (SPARQL 1.1 Graph Store HTTP Protocol5). Esta biblioteca también facilita la realización de consultas SPARQL a un triplestore6 (tipo de base de datos que se construye con el propósito de almacenar y recuperar información que ha sido expresada en formato RDF) vía HTTP utilizando la clase EasyRdf Sparql Client, de este modo las consultas realizadas al triplestore retornarán los resultados como objetos PHP.

JQuery7, es un marco de trabajo para el desarrollo de aplicaciones JavaScript, es decir una biblioteca de código que contiene procesos y rutinas ya listos para utilizar, que se ha comprobado que funcionan y por tanto no es necesario

[5] http://www.w3.org/TR/sparql11-http-rdf-update/

[6] http://www.w3.org/2001/sw/RDFCore/ntriples/

[7] http://jquery.com/

reprogramarlos, solo se reutilizan. JQuery permite manejar con facilidad las peticiones asincrónicas realizadas a un servidor Web a través de AJAX.

OpenLink Virtuoso Universal Server8 es una solución de almacén híbrido para un rango de modelos de datos, incluye datos relacionales, RDF y XML, así como documentos de texto libres. Ofrece además dos variantes para su uso, la primera es Virtuoso Open Source Edition (es la que mayormente utilizan los desarrolladores puesto que es de código abierto) y la otra es una edición comercial, que requiere el pago de una licencia para su uso.

Virtuoso ofrece directamente un SPARQL Endpoint9 que permite la consulta de los recursos contenidos en dicho servidor, proporciona además una herramienta de gestión mediante una interfaz Web llamada Virtuoso Conductor, desde la que se tiene completo acceso a todas las funcionalidades disponibles.

El desarrollo de la propuesta de solución ha sido realizado siguiendo un enfoque ágil de desarrollo de software, utilizando la metodología Programación Extrema (XP, por sus siglas en inglés).

El objetivo fundamental de utilizar XP10 viene dado por una serie de características particulares de esta metodología que resultan aplicables durante el proceso de desarrollo de la propuesta de solución. En primer lugar, la idea de producir rápidamente versiones del sistema a desarrollar, que sean operativas aunque obviamente no puedan contar con toda la funcionalidad pretendida para el sistema sí es necesario que constituyan un resultado de valor para el negocio. Por otro lado se cuenta con la disponibilidad permanente del cliente, el cual forma parte del equipo de desarrollo. Las tareas de programación se realizan por parejas de programadores, priorizando así la comunicación interpersonal y el flujo constante de información entre los programadores. Otra característica importante de XP es que esta se define como especialmente adecuada para proyectos pequeños con requisitos imprecisos y cambiantes, y donde existe un alto riesgo técnico (K Beck 1999).

[8] http://virtuoso.openlinksw.com

[9] SPARQL *Endpoint*, o punto de acceso a base de datos, es la interfaz que permite realizar consultas a un triplestore

[10] www.extremeprogramming.org

Con el propósito de construir interfaces intuitivas, amigables y que satisfagan los criterios básicos de usabilidad para aplicaciones Web, la información que se muestra al usuario ha sido estructurada en base a patrones de interacción comunes en la Web y que pueden ser fácilmente extendidos al contexto de los datos enlazados. A continuación se muestran las principales tareas utilizadas para el análisis del conjunto de datos junto al patrón interactivo que la satisface y el componente de la AI que implementa dicho patrón.

Filtrar los elementos de interés, y obviar aquellos que no son de interés para el usuario: Aquí la propuesta es utilizar el patrón interactivo Navegación Facetada11, las facetas son el componente de la AI que permite a los usuarios filtrar los elementos dentro del conjunto de datos, obviando los elementos que no son de su interés. Con esta estrategia se reduce el espacio de búsqueda por cada restricción aplicada hasta que se obtiene el elemento o elementos de interés.

Mostrar los detalles de los recursos de interés: Una vez que el usuario ha obtenido el (los) elemento(os) que requiere a través del patrón interactivo Navegación Facetada, es necesario que se muestre información detallada del elemento, esto en el contexto de los datos enlazados se reduce a listar el conjunto de propiedades y sus respectivos valores para cada uno de los elementos filtrados. Aquí se aplica el patrón interactivo Detalles sobre la Demanda12 y el componente de la AI que lo implementa es el menú desplegable mostrando las propiedades y valores de cada elemento filtrado.

Contextualizar el espacio de navegación: Cuando el usuario “navega” el conjunto de datos a través de las facetas o a través de otro tipo de búsqueda (búsqueda textual) como se verá posteriormente, puede ocurrir que este requiera orientarse y saber en qué parte de la búsqueda se encuentra y hacia dónde puede llegar desde su ubicación actual. Con este propósito se aplica el patrón interactivo Breadcrumbs13 , este ofrece un punto de referencia para el usuario durante toda la navegación y los procesos de búsqueda, así sabrá en cualquier momento en qué parte de la búsqueda se encuentra y dispondrá de enlaces directos a posiciones anteriores del camino recorrido. El componente de la arquitectura de la

[11] http://www.welie.com/patterns/showPattern.php?patternID=faceted-navigation

[12] http://www.welie.com/patterns/showPattern.php?patternID=details-on-demand

[13] http://www.welie.com/patterns/showPattern.php?patternID=crumbs

información que implementa este patrón interactivo recibe el nombre de breadcrumbs (equivalente a "migajas de pan" en Español).

Sugerir opciones presentes en el conjunto de datos a medida que el usuario introduce el criterio de búsqueda: A los efectos de la búsqueda textual, resulta útil al usuario que a medida que se introduce algún criterio de búsqueda, se brinden un conjunto de opciones que coincidan con el criterio que se va introduciendo. Con esta finalidad se aplica el patrón interactivo Autocomplete14 (equivalente a Auto completamiento en Español), el cual puede ser implementado a través de componentes de la AI tales como radios, y combobox.

3. RESULTADOS

La propuesta de solución tiene como punto de partida, la implementación de las tareas y patrones de interacción definidos en la sección 2. Estos patrones son comunes en aplicaciones Web y se extienden al contexto de los datos enlazados brindando oportunidades para que los usuarios no técnicos realicen un consumo intuitivo de los datos.

La usabilidad se encuentra determinada por la aplicación de estos patrones de interacción a través de los componentes de la AI que lo implementan. Esto está estrechamente relacionado con la caracterización del conjunto de datos con el que se trabaja, lo que se reduce en mostrar al usuario los recursos, propiedades e interrelaciones existentes entre los datos, así como brindar opciones de navegabilidad a través de ellos. Para caracterizar el conjunto de datos se hace necesario interactuar con la estructura de datos subyacente, el grafo RDF que se encuentra almacenado en el *triplestore* del servidor Virtuoso.

Las solicitudes de información realizadas por los usuarios al interactuar con los componentes de la AI, son convertidas dinámicamente en peticiones asíncronas parametrizadas que se envían al *Endpoint* que proporciona el servidor Virtuoso a través del protocolo HTTP. Los parámetros enviados en las peticiones incluyen la consulta a ejecutar sobre el grafo RDF (utilizando la sintaxis de SPARQL), así como el formato en que se requiere obtener el resultado (HTML, JSON, XML, etc.).

[14] http://www.welie.com/patterns/showPattern.php?patternID=autocomplete

Los resultados se muestran dinámicamente en cada uno de los componentes de la interfaz HTML con la que interactúa el usuario según los procesos descritos en las secciones 3.2.1 y 3.2.2. Mientras la sintaxis RDF de los datos está completamente oculta para el usuario, la navegación a través del grafo RDF es posible porque ocurre la traducción de las solicitudes del usuario a consultas SPARQL ejecutadas sobre el *Endpoint* del servidor Virtuoso.

3.1. Componentes de la Arquitectura de la Información

Cuando un usuario interactúa con un conjunto de datos, necesita mecanismos que le permitan explorar y conocer la estructura de los mismos. En la sección 2, se abordó cómo a través de SPARQL son consultados los datos del conjunto de datos, sin embargo esto es posible solo para los usuarios que conocen el estándar. Por tanto es necesario ofrecer vías alternativas para que usuarios no técnicos también puedan consumir datos enlazados sin tener en cuenta cómo se estructura el dato.

Si evaluamos los mecanismos que son utilizados en la Web para que los usuarios comprendan la estructura de la datos que están navegando, quizás la referencia más importante sea brindada por la disciplina AI (P. Morville, L. Rosenfeld 2006), debido a la gran experiencia acumulada referente a los mecanismos para estructurar los sitios Web y facilitar el acceso de los usuarios a su contenido.

En el contexto de los datos enlazados el uso de los componentes de la AI tiene potenciales aplicaciones, dadas por la forma en que se encuentra estructurado el dato. En el caso particular del conjunto de datos bibliográficos con el que se opera en el proyecto DBJournal, los recursos y sus propiedades son presentados al usuario a través de un componente de la AI denominado faceta.

Las facetas permiten caracterizar los elementos representativos del conjunto de datos y organizarlos de manera que sean comprendidos por los usuarios, ver Fig. 2. Otra aplicación importante de este componente se pone de manifiesto en la búsqueda de resultados filtrando el conjunto de datos a través de restricciones dinámicas, lo que también se conoce como navegación o búsqueda facetada (2011) y se aborda en la sección 3.2.2.

Fig 2. Ejemplo de Facetas: Autor, Fecha, Fuentes

Otro elemento de vital importancia, más aún si se explora un conjunto de datos por primera vez, es el mecanismo que se utiliza para contextualizar la navegación. Como se mostró en la sección 1.1, el componente de la AI *breadcrumbs* proporciona un historial de la navegación del usuario a través del conjunto de datos, este se muestra en la parte superior de la interfaz HTML. Cada segmento de esta cadena se corresponde con un paso del proceso de navegación que coincide con algún recurso o propiedad del conjunto de datos visitado con anterioridad y a la cual se puede acceder de manera directa a través de un clic sobre sobre su nombre.

Un conjunto de datos puede ser explorado también a través de la búsqueda de entidades, haciendo coincidir una cadena de texto específica con el valor de una propiedad que pertenece a algún recurso, esto se conoce como búsqueda textual y se aborda en la sección 3.2.1. En este caso es de utilidad el uso de componentes que ayuden al usuario a encontrar lo que busca, haciendo sugerencias de las opciones de coincidencias de acuerdo al criterio que va introduciendo. Más de un componente visual permite realizar esta tarea, entre ellos: Cajas de textos y listas.

Siempre que se tiene un recurso se debe mostrar información que lo describa. En el contexto del conjunto de datos bibliográficos del proyecto DBJournal, si se muestran los artículos de cierto autor, debe visualizarse información que caracterice al artículo, tales como: título, autor(es), coautor(es), entre otros elementos descriptivos. Lo anterior se aplica en la propuesta de solución a través de la implementación del paradigma: Detalles sobre la demanda (abordado en la sección 1.1), el cual es implementado mediante el uso de componentes visuales como los *div* (capas), los cuales son generados dinámicamente a partir de los resultados de las consultas SPARQL.

3.2. Paradigmas de Búsqueda

Para los usuarios no técnicos que desconocen la estructura y forma de consultar los datos, resulta indispensable contar con mecanismos que permitan la búsqueda de información sin un conocimiento previo de estos dos elementos. Por tanto, se hace necesario desarrollar interfaces con soporte para realizar búsqueda intuitiva, que sean además fáciles de usar y sobre todo capaces de satisfacer las necesidades de información de los usuarios que interactúan con ellas. Una solución a este problema es la inclusión de funcionalidades basadas en paradigmas de búsqueda en las interfaces de usuario.

En la concepción tradicional de la Web (Web de los documentos) se utilizan paradigmas de búsqueda, cuya aplicación ha sido extendida al contexto de los datos enlazados. Los paradigmas de búsqueda existentes se clasifican en tres categorías: Palabras Clave, Iterativo-Exploratorio, Lenguaje Natural (1962).

3.2.1. Búsqueda Textual

La formulación de necesidades de información a través de palabras claves, es un paradigma que está siendo aplicado a las aproximaciones de búsqueda en la Web Semántica que operan con datos enlazados. Muchos motores de búsqueda semánticos realizan búsqueda de entidades, por ejemplo Falcons (2009) y Sig.ma (2010) proporcionan una interfaz que responde a necesidades de información de los usuarios a través de la búsqueda por palabras claves.

Este paradigma ha sido implementado también en motores de búsqueda como SemSearhPro e IBM’s Avatar. Estos permiten satisfacer necesidades de información complejas al relacionar entre sí las entidades obtenidas mediante la búsqueda por palabras clave.

La ventaja principal de este paradigma está dada en que permite utilizar mecanismos de indexación, lo que hace que solo sea necesario indexar una sola vez, y luego consultar el índice cada vez que se realice una consulta, esto agiliza y simplifica el proceso de búsqueda.

No existe una sintaxis estándar para expresar consultas de búsqueda textual en SPARQL (solo es posible utilizar expresiones regulares para buscar coincidencias con cadenas de texto en objetos del grafo RDF). Sin embargo, cada proveedor de almacén de tripletas RDF (*triplestore*) proporciona extensiones específicas para

la búsqueda textual (Virtuoso[15], LARQ[16], Lucene Sail[17]). Por tanto es posible realizar consultas híbridas que combinan SPARQL y búsqueda textual, estas consultas arrojan mejores resultados que los obtenidos por consultas expresadas en SPARQL utilizando expresiones regulares para soportar la búsqueda textual sobre RDF (2011).

El servidor Virtuoso posibilita la creación de índices a partir del grafo RDF almacenado en su *triplestore*, una vez construido un índice se crea un nuevo espacio de nombres denominado ***bif***[18]**,** a través del cual es posible consultar directamente el índice utilizando el procedimiento almacenado ***contains*** en consultas SPARQL híbridas. La propuesta de solución en desarrollo utiliza el mecanismo anterior para implementar el paradigma de la búsqueda textual.

La idea es transformar el criterio de búsqueda en consultas SPARQL parametrizadas utilizando el predicado **bif: contains(`[expresión regular]),** también es posible de manera opcional utilizar expresiones regulares como argumento para buscar coincidencias con las entradas del índice.

El paradigma de búsqueda a través de palabras claves presenta dos desventajas fundamentales, la primera de ellas es la posible ocurrencia de ambigüedades en las búsquedas (cuando una palabra tiene más de un significado).

La segunda desventaja radica en que este paradigma asume un escenario de búsqueda donde el usuario tiene precisión en sus necesidades de información, de modo que requiere de un conocimiento previo del dominio de interés para la búsqueda. Esto puede resultar un obstáculo en las situaciones más comunes, es decir en escenarios donde las necesidades de información no son precisas o están vagamente definidas (el usuario desconoce con certeza qué debe buscar). En estos casos es conveniente utilizar el paradigma iterativo/exploratorio.

El paradigma iterativo/exploratorio permite realizar una búsqueda iterativa mediante la navegación exploratoria de un conjunto de datos. La navegación puede ser realizada mediante facetas o a través de la visualización de grafos.

[15] http://docs.openlinksw.com/virtuoso/sparqlextensions.html
[16] http://jena.sourceforge.net/ARQ/lucene-arq.html
[17] http://dev.nepomuk.semanticdesktop.org/wiki/LuceneSail
[18] http://docs.openlinksw.com/virtuoso/sparqlextensions.html#rdfsparqlrulefulltext

De manera particular la propuesta solución implementa la variante de la navegación facetada (búsqueda facetada).

3.2.2. Búsqueda Facetada

La navegación facetada constituye una técnica para la exploración de datos estructurados basada en la teoría de la faceta (1962), esta técnica permite explorar conjuntos de datos a través de dimensiones conceptuales ortogonales, también llamadas facetas, las cuales no son más que representaciones de las características importantes de los elementos o recursos.

Por otro lado la principal ventaja de la navegación facetada radica en las facilidades que esta ofrece al responder a necesidades de información en un escenario de búsqueda donde el usuario no tiene claro qué desea buscar, ni cómo debe hacerlo (escenario de búsqueda vago). Además no es necesario un conocimiento a priori del esquema de los datos, ya que al implementar el paradigma exploratorio, las necesidades de información del usuario se satisfacen a través de la exploración (navegación) del conjunto de datos. Otra ventaja de la navegación facetada es que evita la ambigüedad y las consultas se construyen de forma estructurada, utilizando el lenguaje de consultas.

La propuesta de solución implementa este paradigma de búsqueda a través de consultas SPARQL parametrizadas e interacciones AJAX complejas en JQuery. A medida que el usuario adiciona restricciones mediante la selección de valores en las facetas, se generan consultas SPARQL que utilizan estos valores como parámetros para la construcción del patrón de tripletas de la consulta correspondiente.

5. CONCLUSIONES

Como se ha mostrado, la propuesta de solución en desarrollo brinda un conjunto de características que permiten al usuario no técnico consumir los datos bibliográficos que han sido publicados como datos enlazados por el proyecto DBJournal. Lo anterior tiene como base los patrones de interacción aplicados, así como los componentes de la arquitectura de la información que los implementan y que proporcionan mayor intuitividad e interactividad con el usuario.

En la propuesta de solución en desarrollo, la búsqueda textual satisface las necesidades de información del usuario haciendo coincidir un criterio de búsqueda

basado en texto con las entradas de un índice generado a partir de los objetos de un grafo RDF; este índice es consultado vía SPARQL. Sin embargo presenta desventajas tales como la posible ambigüedad en el criterio para la búsqueda. Por otro lado requiere que el usuario conozca el dominio sobre el cual debe buscar, lo que no ocurre con frecuencia en la práctica, por tanto no es útil en escenarios de búsqueda poco precisos. Es por ello que se ha incorporado además la búsqueda facetada. La combinación de ambos paradigmas de búsqueda ofrece al usuario mayores probabilidades de satisfacer sus necesidades de información.

Es posible además una caracterización de los recursos y propiedades del conjunto de datos a través de las facetas generadas. Estas muestran los recursos, la cantidad de instancias por recurso que existen en el conjunto de datos y ofrecen oportunidades de navegación a través de ellos (navegación facetada).

Aunque son soportadas las funcionalidades descritas anteriormente, aún es necesario continuar el trabajo en el mecanismo de generación de las facetas.

Las facetas generadas por la aplicación en desarrollo son dependientes de las ontologías con las que se describen los datos bibliográficos en el conjunto de datos. De modo que si se adicionan nuevas ontologías para describir nuevas clases dentro de los datos bibliográficos sería necesario redefinir las consultas SPARQL que recuperan los recursos y propiedades desde el grafo RDF almacenado en el servidor.

En (Roberto García, Josep Maria Brunetti, Antonio López-Muzás, Juan Manuel Gimeno, Rosa Gil 2013) se propone una solución a este problema, sin embargo, está basada en la extracción y procesamiento de las clases del conjunto de datos para generar las facetas, lo cual no resulta de utilidad en el contexto de la propuesta de solución en desarrollo donde la generación de las facetas ocurre a partir de las propiedades de los recursos del conjunto de datos.

6. REFERENCIAS BIBLIOGRÁFICAS

Aba-Sah Dadzie, & Matthew Rowe. (2011a). APPROACHES TO VISUALISING LINKED DATA: A SURVEY. *IOS Press*, *2*(2), 89-124. http://doi.org/10.3233/SW-2011-0037

Benjamin Nowack. (2009). PAGGR: LINKED DATA WIDGETS AND DASHBOARDS. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, *7*(4), 272 – 277.

Eric Prud'Hommeaux. (2008a). SPARQL QUERY LANGUAGE FOR RDF. W3C. Recuperado a partir de http://www.w3.org/TR/rdf-sparql-query/.

Fadi Maali, Richard Cyganiak, & Vassilios Peristeras. (2011). RE-USING COOL URIs: ENTITY RECONCILIATION AGAINST LOD HUBS (Vol. 8). Presentado en Proceedings of the Linked Data on the Web Workshop. Recuperado a partir de : http://ceur-ws.org/Vol-813/ldow2011-paper11.pdf.

G. Anuradha, G.Sudeepthi, & Prof. M.Surendra Prasad Babu. (2012). A SURVEY ON SEMANTIC WEB SEARCH ENGINE. *IJCSI International Journal of Computer Science Issues*, *9*(2), 1-5.

Gong Cheng, & Yuzhong Qu. (2009). SEARCHING LINKED OBJECTS WITH FALCONS: APPROACH, IMPLEMENTATION AND EVALUATION. *Journal of Semantics Web*, *5*(3), 49–70.

Michele Catasta, & Giovanni Tummarello. (2010). SIG.MA: LIVE VIEWS ON THE WEB OF DATA. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 355 – 364. http://doi.org/10.1145/1772690.1772907

Rafael Rodríguez Fuentes, Y. H. D. (2013). La Web Semántica: Una Breve Revisión. *Revista Cubana de Ciencias Informáticas*, *7*(1).

Roberto García, Jose Maria Brunetti, Antonio López-Muzás, Juan Manuel Gimeno, & Rosa Gil. (2011). Publishing and Interacting with linked Data. Presentado en Proceedings of the International Conference on Web Intelligence, Sogndal, Norway. http://doi.org/10.1145/1988688.1988710

Shiyali R. Ranganathan. (1962). *ELEMENTS OF LIBRARY CLASSIFICATION* (3rd edition). Bombay: Asia Publishing House.

Thomas R. Gruber. (1993). A translation approach to portable ontology specifications. *ScienceDirect*, *5*(2), 199-220.

Timothy Berners-Lee. (2001). The Semantic Web. *Scientific American Magazine*, 29-37.

TOM HEATH. (2008b). HOW WILL WE INTERACT WITH THE WEB OF DATA ? *Journal IEEE Internet Computing*, *12*(5), 81-85. http://doi.org/10.1109/MIC.2008.101

Tom Heath, & Christian Bizer. (2011b). *LINKED DATA EVOLVING THE WEB INTO A GLOBAL DATA SPACE* (1st edition). Morgan & Claypool Publishers. Recuperado a partir de www.morganclaypool.com