UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

Grupo de Investigación de Web Semántica

"Método para la integración de datos heterogéneos mediante una ontología de dominio."

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

Autor:
Dasiel Miguel Pedrero Mesa

Tutor(es):
Ing. Alejandro Jesús Mariño Molerio
MsC. Yusniel Hidalgo Delgado

La Habana, junio de 2018
"Año 60 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 _____.

_____ _____

Dasiel Miguel Pedrero Mesa Ing. Alejandro Jesús Mariño Molerio

Autor Tutor

_____

MsC. Yusniel Hidalgo Delgado

Tutor

DATOS DE CONTACTO

Síntesis del Tutor

El Ing. Alejandro Jesús Mariño Molerio se graduó en la Universidad de Ciencias Informáticas en el año 2016. En su primer año de adiestramiento ocupó el rol de coordinador en el Grupo de Investigación de Web Semántica. Actualmente se desempeña como profesor del departamento docente de técnicas de programación de la Facultad 3.

El máster en ciencias Yusniel Hidalgo Delgado se graduó con Título de Oro en la Universidad de 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 auxiliar 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.

*A **Reina**, **Dayman** y **Meña** que ya no están en este mundo.*

*A mi hermana **Lissett**.*

*A mis hermanos **Danger** y **Mayckell**.*

*A mi **madre**.*

Gracias a mis tutores por su ejemplo de profesionalidad y la confianza depositada en mí.

Gracias a todos mis amigos, los de afuera, los de adentro y los del fútbol. En especial a Raykof, Alexis, Cesar por entregarme su amistad y por ser incondicionales a ella. A Ingrid mi mai, mi mejor amiga, por ayudarme a levantarme tantas veces, por aconsejarme, a ti muchas gracias. A Nailee por convertirse en mi consejera, por enseñarme a ver la vida en otro sentido con tus pensamientos. A Luis, Víctor, Osvaldo, Dalber, Jennifer, Oscar y Lucio, por ayudarme en todo lo que pudieron y compartir conmigo momentos inolvidables.

Gracias a mi amigo Jesús por ser mi hermano, mi mano derecha, mi compañero de mil batallas y siempre estar ahí conmigo hasta el final.

Gracias a mi hermana Lissett por ser ejemplo dentro de la casa, por ser dulce y cálida, aunque no se diera cuenta, por hacer de nuestras diferencias el eslabón que nos uniera, a ti muchas gracias. A mis hermanos Danger y Mayckell por ser mis guías, mis referencias paternales, por ser buenos hermanos y hombres de bien.

Gracias a mi novia Daniela, por su amor incondicional, por estar conmigo en las buenas y en las malas, por aguantar mis berrinches, por su paciencia.

Gracias a la persona que siempre confío en mí cuando nadie lo hizo, a la que apostó por mí cuando todos le dijeron que no lo hiciera, al amor de mi vida, a la mujer que fue padre, hermana, y sobre todo madre por encima de todas las cosas. A ti mami muchas gracias, tu eres la responsable de que esto hoy fuera posible, muchísimas gracias.

RESUMEN

La integración de datos es el proceso que permite combinar datos heterogéneos de múltiples fuentes en la forma y estructura de una única aplicación. Este proceso de integración de datos facilita que diferentes tipos de datos, tales como matrices de datos, documentos y tablas, sean fusionados por usuarios, organizaciones y aplicaciones para un uso personal, de procesos de negocio o de funciones. Dentro de la variedad de datos que son generados en el mundo, se encuentran los que son emitidos por instituciones del tipo gubernamental, estos toman un valor significativo y son llamados datos abiertos gubernamentales. Estos datos pueden reflejar el comportamiento de una entidad y su misión es que los usuarios puedan acceder a ellos de manera abierta y sin costo alguno. El grupo de investigación de web semántica de la Universidad de las Ciencias Informáticas desarrolla un proyecto de investigación con el objetivo de procesar y publicar datos institucionales como datos abiertos gubernamentales. Este proceso de integración de datos permitirá un correcto uso de la información de carácter gubernamental para la toma de decisiones, acceso o modificación de los mismos. La presente investigación propone un método que integra datos gubernamentales a una ontología de dominio.

Palabras claves: integración de datos, datos gubernamentales, ontología, web semántica.

ABSTRACT

Data integration is the process that allows you to combine heterogeneous data from multiple sources in the form and structure of a single application. This process of data integration facilitates that different types of data, such as data matrices, documents and tables, be merged by users, organizations and applications for personal use, business processes or functions. Within the variety of data that are generated in the world, are those that are issued by government-type institutions, these take a significant value and are called open government data. This data may reflect the behavior of an entity and its mission is that users can access them openly and at no cost. The semantic web research group of the University of Information Sciences develops a research project with the objective of processing and publishing institutional data such as government open data. This process of data integration will allow the correct use of government information for decision making, access or modification thereof. The present research proposes a method that integrates government data to a domain ontology.

Keywords: *data integration, government data, ontology, semantic web.*

ÍNDICE

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La Web Semántica es una disciplina científica cuyo objetivo es hacer que las páginas que la forman dejen de ser simples cadenas de caracteres para los ordenadores y se conviertan en textos con sentido, es decir, con semántica, que sean entendibles al igual que lo son para los seres humanos. Tim Berners-Lee planteó en (Berners-Lee, Hendler y Lassila 2001) que la Web Semántica es una nueva forma de contenido web, tal que las computadoras comenzarán una nueva revolución de posibilidades.

En la actualidad el derecho de acceso a la información está estrechamente relacionado con el derecho a la documentación pública. Por estos días el acceso a dicha información por parte de los ciudadanos ha progresado significativamente. Afirma (Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee 2007) que la facilidad de flujo de información es la bendición y a la vez la perdición a gran escala de los sistemas descentralizados existentes en la Web, ya que hay que hacer un correcto uso de información tan poderosa puesta al servicio de todos.

Todos los pueblos tienen derecho al acceso exclusivo y transparente de datos, para su correcto uso, publicación y hasta modificación. Tim Berners-Lee en la Primera Conferencia de Gobierno Abierto en 2007 enunció:" … los datos pertenecen al gobierno, pero deben ser de acceso público…". Esto abrió paso a una nueva etapa en la publicación de los datos, para ello se utilizan tecnologías, estándares y políticas asociadas a la Web Semántica.

Uno de los términos usados son los datos enlazados, que son datos estructurados publicados en la Web siguiendo estándares y buenas prácticas. Según expone (Bizer, Heath y Berners-Lee 2009) los datos enlazados simplemente son una manera de usar la web para crear enlaces entre datos de diferentes fuentes. Estos pueden ser tan diversos como las bases de datos mantenidas por dos organizaciones en diferentes ubicaciones geográficas, o simplemente sistemas heterogéneos dentro de una organización que, históricamente, no han sido fácilmente interoperables a nivel de datos. Técnicamente, los datos enlazados se refieren a datos publicados en la Web de forma tal que son legibles por máquina, su significado está explícitamente definido, está vinculado a otros conjuntos de datos externos y, a su vez, puede vincularse desde conjuntos de datos externos.

En el año 2007 se crea el Linked Open Data Project con la tarea de hacer los datos accesibles a todas las personas. Según expone el portal World Wide Web Consortium[1] el objetivo de este proyecto es hacer que los datos estén disponibles gratuitamente para todos. Existen varios conjuntos de datos abiertos disponibles en la Web. Los ejemplos incluyen Wikipedia, Wikilibros, Geonames, MusicBrainz, WordNet, la base de datos bibliográfica DBLP y muchos más que se publican bajo licencias de Creative Commons o Talis. Como paradigma principal de W3C SWEO [2]Linking Open Data se tiene el ampliar la Web con un fondo común de datos publicando varios conjuntos de datos abiertos como RDF[3] en la Web y estableciendo enlaces RDF entre elementos de datos de diferentes fuentes de datos.

Los enlaces RDF permiten navegar desde un elemento de datos dentro de una fuente de datos a elementos de datos relacionados dentro de otras fuentes usando un navegador web semántico. Los rastreadores RDF también pueden ser seguidos por los rastreadores de los motores de búsqueda de la Web Semántica, que pueden proporcionar sofisticadas capacidades de búsqueda y consulta sobre los datos rastreados. Como los resultados de la consulta son datos estructurados y no solo enlaces a páginas HTML, se pueden usar dentro de otras aplicaciones.

La dualidad más evidente ante los datos enlazados es el equilibrio entre publicar y consumir. En (Saorín 2012) se expone que publicar supone adaptar los sistemas para generar fuentes de datos de calidad, basadas en vocabularios RDF y enlazadas a recursos de otros conjunto de datos. Consumir supone la explotación de los datos externos disponibles para aportar valor a los servicios digitales: combinar datos en una aplicación para el usuario final.

En el marco de este proyecto los datos enlazados han crecido como una de las principales actividades por muchas instituciones del tipo gubernamental, surgiendo lo que se conoce como datos enlazados abiertos gubernamentales (LOGD, por sus siglas en inglés). Con los LOGD se busca la apertura y vinculación de datos gubernamentales usando tecnologías de la web

[1] www.w3.org. El portal World Wide Web Consortium (W3C) es una comunidad internacional donde las organizaciones, miembros, el personal a tiempo completo y el público trabajan juntos para desarrollar estándares web. Liderado por el inventor Web Tim Berners-Lee y el CEO Jeffrey Jaffe, la misión del W3C es llevar a la Web a su máximo potencial.

[2] El Grupo de Interés de Educación y Divulgación de la Web Semántica (SWEO) se estableció para desarrollar estrategias y materiales para aumentar la conciencia entre la comunidad Web de la necesidad y el beneficio de la Web Semántica, y educar a la comunidad Web sobre las soluciones y tecnologías relacionadas.

[3] The Resource Description Framework (**RDF**) es un lenguaje para referenciar la información de los recursos de la World Wide Web.

semántica. Afirma (Janssen, Charalabidis y Zuiderwijk 2012) que el público se convierte en parte del sistema de procesamiento de datos y puede procesar datos, enriquecer datos, combinarlos con otras fuentes e incluso podría recopilar sus propios datos (por ejemplo, a través del uso de sus teléfonos móviles). Esto requiere un cambio en los límites tradicionales entre las organizaciones públicas y el público en el que prácticamente cualquier persona en el mundo tiene acceso a los datos.

Adiciona (Villar, Castro y Bermúdez 2017) que el flujo de información a través del uso de la tecnología y los medios digitales ha favorecido las formas de comunicación y de interacción, esta transformación global ha llegado a todos los ámbitos de la vida y es así como las formas de gobierno no han sido exentas de este cambio transformándose en gobiernos abiertos cuya característica es la amplia utilización de las tecnologías de la información y la comunicación con el fin de generar transparencia, participación y colaboración por medio de la apertura de la información pública, de lo cual se espera el interés activo de la sociedad civil por los asuntos públicos.

En una cultura política de arriba hacia abajo donde el estado es el proveedor de servicios de mayor confiabilidad, el estado se convierte en un poderoso monopolio de datos, capaz de estructurar y homogeneizar las interacciones entre sí mismo y sus ciudadanos. Estas interacciones unilaterales son costosas e insensibles a las necesidades de los ciudadanos. Incluso cuando los gobiernos han expuesto datos reales en tiempo real, no se ha logrado presentar datos a los ciudadanos de maneras innovadoras para crear nuevas corrientes de valor. Los proveedores de servicios privatizados han preservado los monopolios del diseño y provisión de servicios. A medida que la tecnología ha incrementado el poder de los datos al facilitar la vinculación y el intercambio, y los pensadores políticos han adoptado la transparencia y el derecho de los ciudadanos a los datos, esta cultura de arriba hacia abajo está siendo desafiada. Muchos gobiernos ahora liberan grandes cantidades de datos en el dominio público, a menudo de forma gratuita y sin gastos administrativos. Según plantea (De Klerk 2011) esto permite la entrega y el diseño de servicios centrados en el ciudadano y mejora la rendición de cuentas de los servicios públicos, lo que lleva a mejores resultados en el servicio público.

Según (Villar, Castro y Bermúdez 2017) la apertura de datos abiertos gubernamentales (Open Data Governmet) beneficia económica y socialmente a la región, socialmente las aperturas de los datos son fundamentales para propiciar confianza en las entidades estatales, lo que conlleva a afianzar vías de participación, empoderar a la ciudadanía como veedores de la gestión pública, mejorar los servicios públicos, crear nuevas ideas de negocio, dar amplio cubrimiento a noticias del día a día para generar análisis a partir de datos de primera mano, completos y oportunos, y en registros investigativos y académicos que pueden ayudar a conseguir un alcance de tipo científico.

Los conjuntos de datos son a menudo en formato de valores separados por comas (CSV) u hojas de cálculo, pero existe la posibilidad de aumentar su utilidad vinculándolos mediante formatos estructurados procesables por máquina. RDF es el formato más integrado en el pensamiento actual sobre las generaciones futuras de la Web, ya que su uso de URI permite identificar los datos por referencia y vincularlos con otros datos relevantes por tema, predicado u objeto. El uso de los estándares de la Web Semántica en datos de gobierno abierto (DAG) fue impulsado por Advanced Knowledge Technologies (AKT). Esta visión es conocida como la web de datos vinculados (LDW). LDW ya está bien poblada a través de iniciativas como DBpedia[4], DBLP Computer Science Bibliographie[5], London Gazette, New York Times y Comprehensive Knowledge Archive Network (CKAN)[6].

Los formalismos y la infraestructura están apareciendo de acuerdo a los principios de datos enlazados establecidos por Tim Berners-Lee hace algún tiempo, pero todavía es necesario abordar cuestiones vitales de investigación (De Klerk 2011). Para un correcto uso y consumo se transforman conjuntos de datos relacionados con el gobierno en RDF, enlazándolos a la Web de Datos y proporcionando demos y tutoriales sobre el procesamiento y consumo de datos gubernamentales vinculados.

[4] **DBpedia** (http://**dbpedia**.org) es un repositorio en la web, abierto y gratuito, con información estructurada proveniente de Wikipedia (http://www.wikipedia.org). A diferencia de Wikipedia, compuesta por documentos, **DBpedia** es un conjunto de datos estructurados.

[5] **DBLP** es un sitio web de bibliografía informática. Comenzando en 1993 en la Universidad de Trier, Alemania, creció a partir de una pequeña colección de archivos HTML y se convirtió en una organización que alberga una base de datos y un sitio de bibliografía de programación lógica.

[6] **Comprehensive Knowledge Archive Network** (**CKAN**) es una aplicación web de código abierto para el almacenamiento y la distribución de los datos, tales como hojas de cálculo y los contenidos de las bases de datos. Está inspirado en las capacidades de gestión de paquetes comunes para abrir sistemas operativos, como Linux, y está destinado a ser el "apt-get de Debian para los datos.

En Cuba, se ha investigado acerca de la importancia del trabajo con datos abiertos, sin embargo, el uso de los datos enlazados abiertos gubernamentales y políticas asociadas aún es pobre, al no contar con un sitio capaz de proporcionar a la población datos gubernamentales en tiempo real. No obstante, se cuenta con los datos publicados en los sitios de instituciones, ministerios y empresas, como por ejemplo la Oficina Nacional de Estadística e Información (ONEI). Estos datos se encuentran dispersos y en diferentes formatos impidiendo establecer relaciones entre los datos de diferentes instituciones a pesar de pertenecer al mismo dominio, el de los datos gubernamentales. Por otra parte, en la Universidad de las Ciencias Informáticas, se desarrolla un proyecto para la publicación de estos datos utilizando estándares y buenas prácticas para la publicación de los datos institucionales como datos abiertos gubernamentales.

Una de las fases se define como la extracción e integración de los datos institucionales publicados a una ontología de dominio. Según aborda (Gillani y Ko 2015) las ontologías se han estudiado durante mucho tiempo en los campos de las tecnologías semánticas, la inteligencia artificial y la gestión del conocimiento.

En este contexto es necesario transformar los datos publicados por las instituciones nacionales, en diferentes formatos, para poblar la ontología de dominio DOGD (Domain Ontology for e-Goverment Data). Cuando se habla de poblar una ontología se refiere al proceso de introducir la información del modelo relacional o cualquier modelo al modelo ontológico. Las tuplas almacenadas en la base de datos se convierten en instancias formando el ABox de las ontologías. Según (Gillani y Ko 2015) con el poblado de una ontología de dominio se busca:

• Combinar medidas de relación semántica y técnicas de adquisición de patrones para la extracción de información.
• Población y enriquecimiento de ontologías de dominio de gran escala que pueden facilitar el proceso de enriquecimiento manual de ontologías que consume mucho tiempo.

Todo lo expuesto hasta el momento hace reflexionar que esta situación constituye un problema de integración de datos, al no contar con un proceso estandarizado para este propósito. La imposibilidad de acceder a datos emitidos en diferentes documentos, formatos, estructuras

puede traer consigo consecuencias no deseadas para cualquier empresa o gobierno electrónico como, por ejemplo:

- Falta de confianza en la información.
- Imposibilidad de tomar decisiones críticas a partir de los datos por no ser fiables. Al encontrarse dispersos, es un proceso agotador el buscar documentos con los datos deseados para tomar una decisión, resulta más fácil tenerlos integrados en un mismo sistema.
- No poder acceder a los datos en el momento en el que se necesitan desde cualquier lugar.
- Carecer de información en tiempo real.

Al realizar un análisis de la situación planteada se define como **problema a resolver** el siguiente ¿Cómo integrar fuentes de datos gubernamentales en una ontología de dominio en un tiempo razonable?

Se entiende por tiempo razonable la razón de analizar y procesar un documento por segundo.

Se define como **objeto de estudio** de la presente investigación la integración de datos, quedando enmarcado como **campo de acción** las fuentes de datos heterogéneas.

Como **objetivo general** se definió:

Desarrollar un método para la integración de datos heterogéneos mediante el poblado de una ontología de dominio para la toma de decisiones en el contexto de los datos gubernamentales.

Para asegurar el cumplimiento del objetivo general 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 método para la integración de datos gubernamentales utilizando una ontología de dominio.
3. Implementar un método para la integración de datos gubernamentales a una ontología de dominio.
4. Validar los resultados obtenidos con la utilización del componente de software

desarrollado mediante la realización de un diseño experimental.

Como posible **resultado a entregar** se define:

Código fuente de un método para la extracción e integración de datos a la ontología de dominio DOGD.

Se tiene como **idea a defender**:

Si se desarrolla un método para la extracción e integración de datos gubernamentales a una ontología de dominio en tiempo razonable, entonces se reducirá el tiempo empleado para acceder a los datos publicados por diferentes instituciones.

Durante la investigación se han empleado un conjunto de métodos científicos para el análisis y obtención de la información relacionada con el objeto de estudio y campo de acción de la investigación, los cuales se explican a continuación.

Métodos Teóricos

Con la aplicación del método Analítico-Sintético, se analizaron cada uno de los conceptos, que de una forma u otra guardan relación con el objeto de estudio de la investigación y las relaciones existentes entre ellos. Con los resultados de este análisis se realiza una síntesis para comprender las características del objeto de estudio y sus relaciones. Este método permitió definir el marco teórico de la investigación propuesto en el Capítulo 1.

Se utilizó el método Inductivo-Deductivo en la confección del estado del arte de la investigación. Este método se basa en los procedimientos inducción y deducción, donde el primero permite arribar a conclusiones generales a partir del estudio de las aproximaciones existentes sobre el objeto de la investigación y el segundo posibilita a partir de un razonamiento lógico sobre lo anterior inferir nuevos conocimientos que lleven al planteamiento de una posible solución para la investigación.

Métodos Empíricos

Entre los métodos empíricos existentes se seleccionaron para su utilización el de Medición y el Experimental, el primero para obtener información numérica acerca del tiempo que se

demora en realizar la extracción e integración de datos a una ontología de manera manual y el obtenido luego de aplicar el mismo procedimiento con la propuesta de solución. Para la validación de la solución propuesta se hace uso del método Experimental, tomando como muestra los datos obtenidos en la medición.

El presente trabajo de diploma consta de una introducción y de tres capítulos, a continuación, un resumen de cada uno de ellos:

Capítulo 1. Fundamentación teórica: se presentan los elementos teóricos que sirven de base a la investigación del problema planteado, analizando los principales conceptos y definiendo en que consiste la relación entre los datos publicados por instituciones gubernamentales y la ontología DOGD.

Capítulo 2. Descripción de la propuesta: se realiza el diseño y desarrollo de un método para la integración de datos gubernamentales a una ontología de dominio DOGD.

Capítulo 3. Validación de la propuesta: se desarrolla un caso de estudio introduciéndole a la propuesta de solución, información brindada por la Organización Nacional de Estadística e Información (ONEI) y realizando un pre-experimento para comprobar si el método es capaz de integrar esos datos a una ontología de dominio DOGD en un menor tiempo de ejecución que la implementación del método sin la propuesta de solución.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

En la actualidad debido a la inmensidad de datos que se generan por instituciones gubernamentales resulta engorroso poder hacer uso de todos ellos para la toma de decisiones o simplemente ser consultados. Para dar solución a la problemática en cuestión es necesario una investigación en las bibliografías relacionadas con el tema.

En este capítulo, se realiza un análisis de la literatura consultada (epígrafe 1.2), al tiempo que se presenta una taxonomía de los conceptos fundamentales asociados a la investigación (epígrafe 1.3), los cuales permitirán una mejor comprensión de este trabajo. Se muestra un análisis del estado del arte (epígrafe 1.4), donde se trata lo relacionado con el objeto de estudio de la investigación, la integración de datos. Finalmente se exponen los resultados obtenidos luego de realizado un estudio de las aproximaciones más representativas existentes en las fuentes estudiadas (epígrafe 1.5), demostrando que son insuficientes, lo cual justifica el desarrollo de la investigación.

1.2 Análisis bibliométrico y documental.

En la investigación se utilizó la bibliografía asociada al tema, principalmente la publicada en los últimos cinco años. Se consultó diferentes sitios bibliográficos como Google Scholar, IEEE y Springer. De igual manera se visitaron sitios web asociados al tema de investigación como fueron Portal de Datos Abiertos de Chile, Portal de Datos Abiertos del Reino Unido, Portal de Datos Abiertos de Estados Unidos, W3C. A continuación, se muestra en la siguiente tabla un resumen de la bibliografía consultada:

Tabla 1.*Resumen de la revisión bibliográfica consultada.*

Tipo de fuente consultada	**Cantidad consultada**	Cantidad publicada en los últimos 5 años [2014-2018]
Artículos en revista científicas	24	5
Artículos en congresos/ conferencias	8	2

Libros	4	2
Tesis de doctorado	2	1
Páginas web	14	14
Total	53	24

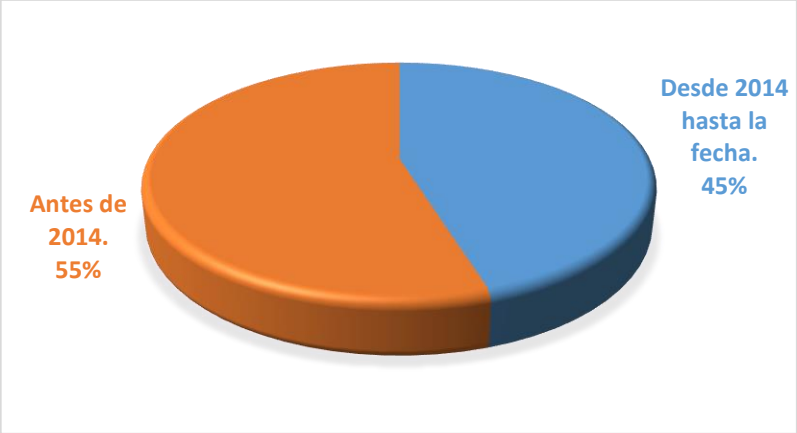

Figura 1. *Actualidad de la bibliografía consultada. (Fuente: elaboración propia).*

1.3 Marco teórico.

En este acápite se describen los principales conceptos de la investigación avalando así la misma, sirviendo como base de conocimiento necesaria para su desarrollo.

1.3.1 Datos Enlazados.

Según (Bizer, Heath y Berners-Lee 2009) el término de datos enlazados se refiere a un conjunto de buenas prácticas para publicar y conectar datos estructurados en la web. Estas buenas prácticas han sido adoptadas luego del incremento del número de proveedores de estos datos en los últimos años, lo que ha llevado a la creación de datos globales, espacio que contiene miles de aserciones: la red de datos.

(Berners-Lee 2006) delineó un conjunto de principios para publicar datos en la Web de forma que todos los datos publicados se vuelvan parte de un único espacio de datos global:

1. Use URI como nombres de la información.
2. Use HTTP URI para que las personas puedan buscar esos nombres.
3. Cuando alguien busca un URI, brinde información útil, usando los estándares (RDF, SPARQL).

4. Incluya enlaces a otros URI, para que puedan descubrir más información.

Asegura(Heath y Bizer 2011) que para entender los principios de los Datos Enlazados es necesario entender la arquitectura del clásico documento web. El documento web se basa en un pequeño conjunto de estándares simples: identificadores uniformes de recursos (URI) como mecanismo de identificación globalmente único, el protocolo de transferencia de hipertexto (HTTP) como mecanismo de acceso universal y el lenguaje de marcado de hipertexto (HTML) como un formato de contenido ampliamente utilizado. Además, la Web se basa en la idea de establecer hipervínculos entre documentos web que pueden residir en diferentes servidores web.

1.3.2 Datos Abiertos.

El constante crecimiento de los datos enlazados impone que más personas deban tener acceso a ellos, ahí es donde entra el término de datos abiertos. La definición de datos abiertos según (Hernández Pérez y García Moreno 2013) recoge once condiciones, para que una obra o un datos sean considerados abiertos: acceso(disponible a un coste razonable y que pueda ser modificable), redistribución, reutilización, ausencia de restricciones tecnológicas, reconocimiento, integridad, sin discriminación de personas o grupos, sin discriminación de ámbitos de trabajo, distribución de licencia, la licencia no debe ser específica de un paquete y la licencia no debe restringir a la distribución de estas obras.

Aporta (Data 2015) que los datos del gobierno se considerarán "abiertos" si los datos se hacen públicos de una manera que cumpla con los siguientes principios:

- Los datos deben estar completos:

Todos los datos están disponibles. El término "datos" se refiere a información o grabaciones almacenadas electrónicamente, incluyendo, pero no limitado a documentos, bases de datos, transcripciones y audio / visual grabaciones. Los datos públicos son datos que no están sujetos a privacidad válida, limitaciones de seguridad o privilegio, como se rige por otros estatutos.

- Los datos deben ser primarios:

Los datos se publican tal como se recopilaron en la fuente con el mejor nivel de granularidad posible, y no en formas agregadas o modificadas.

- Los datos deben estar accesibles:

Los datos están disponibles para la más amplia gama de usuarios para el rango más amplio de propósitos.

- Los datos deben ser procesables por la máquina:
- Los datos están estructurados para que puedan procesarse de manera automatizada.

El acceso debe ser no discriminatorio:

- Los datos están disponibles para cualquier persona, sin requisito de registro.
- Los formatos de datos deben ser no propietarios.

Los datos están disponibles en un formato sobre el cual ninguna entidad tiene exclusividad controlar.

- Los datos deben estar libres de licencia:

Los datos no están sujetos a ningún derecho de autor, patente, marca comercial o comercio regulación de secretos. Privacidad, seguridad y privilegios razonables. Se pueden permitir restricciones según lo regido por otros estatutos.

- Permanencia:

La permanencia se refiere a la capacidad de encontrar información en cualquier momento.

- Costos de uso:

Debe ser completamente gratuito el consumo de los datos abiertos.

Afirma (Janssen, Charalabidis y Zuiderwijk 2012) que los datos abiertos a menudo son indispensables para el desarrollo de políticas públicas y la prestación de servicios. Se define como datos abiertos a información no restringida a la privacidad y datos no confidenciales que se producen con carácter público y están disponible sin ningún tipo de restricciones en su uso o distribución.

Como todo los datos abiertos también tienen su aplicación y (Huijboom y Van den Broek 2011) define cuatro campos primordiales donde los datos abiertos son más útiles, ellos son:
- Educación y capacitación.
- Enfoques voluntarios.

- Instrumentos económicos.
- Legislación y control.

1.3.3 Datos Abiertos Gubernamentales.

Cuando una institución, empresa u organismo emite algún dato, toma un valor significativo, convirtiéndose en datos gubernamentales, si los datos emitidos siguen los principios de los datos descritos anteriormente entonces son datos abiertos gubernamentales. El término “Datos Abiertos Gubernamentales" (OGD) por sus siglas en inglés ha ganado notoriedad hace relativamente poco tiempo, y se popularizó en 2008 luego de la publicación de un conjunto de principios de datos gubernamentales abiertos por parte de los defensores de los derechos humanos en los Estados Unidos.

Según expone (Ubaldi 2013) los dos elementos principales de OGD se definen normalmente de la siguiente manera:

- Datos gubernamentales: son datos e información producidos o encargados por organismos públicos.
- Datos abiertos: son datos que cualquier persona puede utilizar, reutilizar y distribuir libremente, solo sujeto a (como máximo) el requisito de que los usuarios atribuyan los datos y que hagan que su trabajo esté disponible para ser compartido también.

Los datos abiertos gubernamentales están estrechamente relacionados con la información en el sector público (PSI). La PSI es información que fue propiedad y fue publicada por una organización gubernamental y es de libre acceso para cualquiera. Los datos ofrecen más que solo información, lo que es más importante, brinda a los usuarios oportunidad de interpretar los datos ellos mismos (De Klerk 2011).

Resumiendo, los datos abiertos gubernamentales son datos publicados por instituciones del tipo gubernamental. Dichos datos toman valor al ser generados por empresas de alta jerarquía, a la vez esta información servirá para darle al consumidor una visión de cómo son utilizados los recursos y en qué tarea. Además, si una tarea no sale como lo planeado gracias a la publicación de estos datos se sabrá donde existió el problema y será de mayor facilidad darle solución.

1.3.4 Ontología.

En términos de la Web Semántica, cuando se quiere representar algún conocimiento siempre se piensa en una ontología. Una ontología según los filósofos es una teoría que trata de la naturaleza de la existencia, o del tipo de cosas que existen. Los investigadores de la inteligencia artificial y de la red se han apropiado del término y lo han incorporado a su vocabulario; para ellos, una ontología es un archivo o un documento que define formalmente las relaciones entre términos. Es una especificación formal y explicita de una conceptualización compartida (Berners-Lee, Hendler y LASILA 2001). Adiciona (Gruber 1993) una ontología define el vocabulario utilizado para componer expresiones complejas tales como aquellos usados para describir el recurso restricciones en un problema de planificación.

Las ontologías se han convertido en componentes centrales en muchas aplicaciones que incluyen búsqueda, comercio electrónico, configuración y, posiblemente, cada sitio web grande (al menos para organización y navegación). A medida que las ontologías se vuelven más grandes, más distribuidas y más duraderas, la necesidad de los entornos de creación y mantenimiento ontológicos crecen (McGuinness et al. 2000). Una ontología es una forma de representación del conocimiento terminológico en un dominio dado, ayuda a relacionar elementos que pertenecen a un mismo campo.

Según (Abello Diaz 2015) se hace necesario definir un lenguaje para ontologías con tres aspectos importantes:

Conceptualización: El lenguaje debe elegir un modelo de referencia adecuado. Además, debe proporcionar las primitivas correspondientes para representar conocimiento, tales como la definición de entidades y relaciones en un dominio.

Vocabulario: Además de la semántica el lenguaje también debe tener en cuenta la sintaxis, así como la simbología apropiada para indicar los conceptos, así como la gramática para la representación de la conceptualización en una representación explícita.

Axiomatización: Con el fin de capturar la semántica para la inferencia de nuevo conocimiento, es necesaria la implementación de reglas y restricciones, además del conocimiento de hechos.

Adiciona (Abello Diaz 2015) que para compartir conocimiento a través de diferentes dominios se deben tener presentes tres aspectos al desarrollar una ontología:

Extensibilidad: La ontología se debe desarrollar de manera incremental, reutilizando la mayor cantidad de conceptos como sea posible antes de crear un nuevo concepto desde cero.

Visibilidad: El hecho de publicar conocimiento en la Web, no garantiza que éste pueda ser realmente entendido por máquinas y/o humanos. Con el fin de que el conocimiento sea visible en la Web se requieren acuerdos en común sobre la sintaxis y la semántica entre el productor y el consumidor, ya que los agentes de software no son capaces de entender el conocimiento escrito en un lenguaje desconocido.

Inferencia: Una ontología no sólo sirve para la representación del conocimiento de un dominio en particular, sino que también sirve para propósitos de cálculo, ya que permite la inferencia lógica de hechos mediante la axiomatización

1.3.5 Ontología de Dominio.

Las ontologías han dado la vuelta al mundo como formas de representación del conocimiento y sus aplicaciones son diversas. Una de las más usadas son las ontologías de dominio, según (Kaiya y Saeki 2006) una ontología de dominio proporciona una base semántica de requisitos que deben obtenerse. Más concretamente, cada estado que representa un requisito puede ser interpretado con elementos ontológicos y dicha ontología debe incluir conceptos atómicos que cualquiera de las partes interesadas pueden tener comúnmente en un dominio problemático. Una ontología de dominio tiene como característica que para su correcta utilización debe tener implementada acertadamente referencias base primaria del dominio al que pertenece dicha ontología.

1.3.6 Poblar Ontología.

Las ontologías son conocidas por la información que encierran, el proceso de introducirle dicha información a una ontología se conoce como poblar una ontología. Añade (Alani et al. 2003) la población ontológica manual es mano de obra intensiva y consume mucho tiempo.

Algunos enfoques semiautomáticos crean anotaciones de documentos y almacenan los resultados como aserciones en una ontología. Uno de los métodos para poblar una ontología es el de agregar relaciones automáticas entre instancias solo si estas ya existen en la base de conocimientos (ontología).

1.4 Estado del arte.

En este epígrafe se hace un análisis crítico de los temas principales relacionados con la integración de datos y el poblado de ontologías. Para ello se plantean las siguientes incógnitas. ¿Cuáles son las principales herramientas para la extracción de datos? ¿Qué son los wrappers? ¿Qué tipos de heterogeneidades pueden encontrarse? ¿Cuáles son los métodos para el poblado de ontologías más eficientes?

1.4.1 Herramientas para la extracción de datos.

Apache Poi[7]

Apache POI es una API popular que permite a los programadores crear, modificar y mostrar archivos de MS Office utilizando programas de Java. Es una biblioteca de código abierto desarrollada y distribuida por Apache Software Foundation[8] para diseñar o modificar archivos de Microsoft Office utilizando el programa Java. Contiene clases y métodos para decodificar los datos de entrada del usuario o un archivo en documentos de MS Office. La misión del Proyecto POI de Apache es crear y mantener API Java para manipular varios formatos de archivo basados en los estándares Office Open XML (OOXML) y el formato OLE 2 Compound Document (OLE2) de Microsoft. Se puede leer y escribir archivos de MS Excel utilizando Java. Además, puede leer y escribir archivos de MS Word y MS PowerPoint utilizando Java. Apache POI contiene Java Excel (para Excel 97-2008). Es una API completa para portar otros formatos OOXML y OLE2.

Apache Tika[9]

La herramienta Apache Tika detecta y extrae metadatos y texto de más de mil tipos de archivos diferentes (como PPT, XLS y PDF). Todos estos tipos de archivos se pueden analizar

[7] https://poi.apache.org/
[8] https://www.apache.org/
[9] https://tika.apache.org/

a través de una sola interfaz, lo que hace que Tika sea útil para la indexación de motores de búsqueda, el análisis de contenido, la traducción y mucho más.

Internamente, Tika utiliza varios analizadores de documentos existentes y técnicas de detección de tipos de documentos para detectar y extraer datos. Usando Tika, puede desarrollar un detector de tipo universal y un extractor de contenido para extraer tanto el texto estructurado como los metadatos de diferentes tipos de documentos, como hojas de cálculo, documentos de texto, imágenes, archivos PDF e incluso formatos de entrada multimedia hasta cierto punto. Tika proporciona una única API genérica para analizar diferentes formatos de archivo. Utiliza bibliotecas de analizador especializadas existentes para cada tipo de documento. Todas estas bibliotecas de analizador están encapsuladas bajo una interfaz única llamada interfaz del analizador.

Apache Fop[10]

Apache FOP (Formatting Objects Processor) es un formateador de impresión impulsado por objetos de formato XSL (XSL-FO) y un formateador independiente de salida. Es una aplicación Java que lee un árbol de objetos de formato (FO) y procesa las páginas resultantes a un resultado específico. Los formatos de salida soportados actualmente incluyen PDF, PS, PCL, AFP, XML (representación de árbol de área), Print, AWT y PNG, y en menor medida, RTF y TXT. El objetivo de salida primario es PDF.

Al analizar las tres herramientas antes mencionadas, **Apache Tika** destaca por su versatilidad con los formatos con los que puede trabajar. Por las salidas que devuelve, siendo esto la principal ventaja de las otras dos herramientas en cuestión. Además, Apache Tika permite el trabajo con documentos de cualquier versión del paquete Office, mientras que herramientas como el Apache Poi las versiones permitidas están limitadas.

1.4.2 Integración de datos.

Se conoce que la integración de datos es el problema de combinar datos que residen en diferentes fuentes y proporcionar al usuario una vista de estos datos. Según (Lenzerini 2002) el problema de diseñar la integración de datos de sistemas de control es importante en las

[10] https://xmlgraphics.apache.org/fop/

aplicaciones actuales del mundo real, y se caracteriza por una serie de cuestiones que son de interés desde un punto de vista teórico.

Adiciona (Pinkel et al. 2015) que la integración de datos es un gran desafío en la industria, las ciencias de la vida y la web, donde los datos no solo han alcanzado grandes volúmenes, sino que también vienen en una variedad de formatos. La integración de datos aumenta la utilidad de los datos, proporciona un punto de acceso unificado a varias bases de datos y permite analizarlos, por ejemplo, mediante la correlación de sus datos y la identificación de importantes patrones.

Obviamente, una de las tareas principales en el diseño de un sistema de integración de datos es establecer el mapeo entre las fuentes y el esquema global, tal asignación debe ser adecuadamente considerada al formalizar un sistema de integración de datos. Se deduce que los componentes principales de un sistema de integración de datos son el esquema global, las fuentes y el mapeo. El esquema global es una vista unificada de los datos a mostrar al usuario, las fuentes son las direcciones de las cuales se extrajo dichos datos y el mapeo es lo que se realiza entre el esquema global y las fuentes para mostrar al usuario el resultado de su búsqueda. Añaden (Doan, Halevy y Ives 2012) que el objetivo de un sistema de integración de datos es ofrecer un acceso uniforme a un conjunto de fuentes de datos autónomas y heterogéneas.

Además, que existen cuatro puntos a tener en cuenta a la hora de resolver un problema de integración de datos:

Consulta: El enfoque de la mayoría de los sistemas de integración de datos sigue buscando fuentes de datos diferentes. Sin embargo, la actualización de las fuentes es ciertamente de interés.

Número de fuentes: la integración de datos es un desafío para un pequeño número de fuentes (menos de 10 y de más 2). En el extremo, se debe apoyar la integración de datos a escala web.

Heterogeneidad: Un escenario de integración de datos atípicos implica una fuente de datos que se desarrolló de manera independiente entre los demás. Como consecuencia, las fuentes

de recursos operan en diferentes sistemas: algunas de las bases de datos de peso, pero otras pueden ser sistemas de gestión de contenido que simplemente responden al directorio. Los recursos tendrán diferentes esquemas y referencias de los objetos, incluso si ellos moderan esos mismos dominios. Las fuentes de recursos pueden estar completamente estructuradas (por ejemplo, bases de datos relacionales), mientras que otras pueden estructurarse (por ejemplo, XML, texto).

Autonomía: Los recursos no necesariamente se relacionan con una sola entidad administrativa, y si lo hacen, pueden ser utilizados por organizaciones diferentes. Por lo tanto, no se puede asegurar que se tiene acceso completo a la fuente de datos que permite acceder a la información cuando se solicita, y es necesario prestar atención a la privacidad de la información que no corresponda. Además, los recursos pueden cambiar los formatos de datos y los programas de acceso en cualquier momento.

1.4.3 Clasificación de las heterogeneidades.

Según (Anguita Sanchez 2012) los esquemas diseñados para contener datos semánticamente equivalentes son a menudo totalmente incompatibles. A esto se le debe sumar la resolución de las divergencias entre datos debidas a diferentes codificaciones de los mismos. A continuación, se detallan los tipos de heterogeneidades a tener en cuenta:

Heterogeneidades sintácticas

Las heterogeneidades sintácticas son producto de los diferentes métodos y enfoques que se emplean a la hora de gestionar y acceder a las distintas bases de datos. Adiciona (Anguita Sanchez 2012) que existen tres fuentes de heterogeneidades sintácticas:

- Diferencias en los modelos de representación del conocimiento: los datos pueden estar almacenados en un fichero plano (por lo tanto, sin una estructura correspondiente), en una estructura XML, de acuerdo a un modelo relacional, etc. El uso de diferentes modelos de organización imposibilita un homogéneo acceso a los mismos.
- Diferentes lenguajes de consultas: aun cuando dos fuentes de datos comparten la misma forma de representación del conocimiento, es posible que acepten distintos lenguajes de consultas (por ejemplo, SQL, HQL, OQL).
- Tipo de interfaz: la interfaz para acceder a las bases de datos puede ser de distinta naturaleza (Servicios Web, interfaces programáticas, etc).

Heterogeneidades semánticas

Son el resultado de diferencias en la modelización y en la codificación de información semánticamente equivalente. Según (Anguita Sanchez 2012) se dividen en dos categorías:

- Heterogeneidades a nivel de esquema: se refieren a diferencias en la forma de organizar los datos entre diferentes repositorios; es decir, diferencias en los esquemas o en los metadatos.
- Heterogeneidades a nivel de instancia: se refieren a diferencias causadas por el uso de distintas codificaciones para almacenar la información.

1.4.4 Arquitectura básica de un sistema de integración de datos.

Las fuentes de datos pueden ser relacionales, XML o cualquier tienda que contenga datos estructurados. Los envoltorios o cargadores solicitan y analizan los datos de las fuentes. El esquema mediado o el almacén de datos central abstrae todos los datos fuente, y el usuario plantea consultas sobre esto.

Para esto se utilizan los wrappers o envolturas, los wrappers según (Doan, Halevy y Ives 2012) son los componentes de un sistema de integración de datos que se comunican con las fuentes de datos. La tarea del wrapper implica enviar consultas de los niveles superiores del sistema de integración de datos a las fuentes y luego convertirlas a una información que pueda ser manipulada por el procesador de consultas. La complejidad de la envoltura depende de la naturaleza de la fuente de datos. En el caso más simple, si el origen de datos es un sistema de base de datos relacional, entonces la tarea del contenedor es bastante simple y puede implicar simplemente la interacción con un controlador JDBC. En más casos complejos, el wrapper necesita a los servidores de datos estructurados que se encuentran en las páginas HTML y las transforman en un conjunto de tuplas.

Es posible que dos clases no puedan realizar una transferencia de datos debido a la presencia de puntos de acceso a datos incompatibles. Por ejemplo, si una clase acepta una fecha en formato DD / MM / AA de otra clase que genera una fecha en formato MM / DD / AAAA, ninguna de las clases puede coexistir para operar como un módulo integrado porque sus formatos de

datos son diferentes. La clase contenedora resuelve este problema convirtiendo los datos en un formato compatible.

La clave para construir una aplicación de integración de datos son las descripciones de fuente, el pegamento que conecta el esquema mediado y los esquemas de las fuentes. Estas descripciones especifican las propiedades de las fuentes que el sistema necesita saber para usar sus datos.

En el enfoque de almacenamiento, en lugar de un esquema mediado, el usuario plantea consultas en términos del esquema de depósito. Además de ser un esquema que contiene los atributos necesarios de las fuentes, el esquema de depósito también es un esquema físico con una instancia de base de datos subyacente. En lugar de wrappers, el sistema incluye ETL o tuberías de herramienta de extracción-transformación-carga que periódicamente extraen datos de las fuentes y cargan la memoria en el almacén. A diferencia de los wrappers, la ETL aplica, de manera conceptual, una transformación más completa de las estadísticas que puede implicar la limpieza, la agregación y el cambio de formaciones. Estas transformaciones desempeñan el papel de las asignaciones de esquema en la arquitectura de integración de datos virtuales, pero tienden a ser más funcionales.

Algunas de las propiedades del almacenamiento de datos provienen del hecho de que estos sistemas no se desarrollaron originalmente con el propósito de la integración de datos. En cambio, se desarrollaron como una herramienta para realizar un análisis más profundo, como cargar datos de sistemas transaccionales (por ejemplo, bases de datos que registran cada venta realizada en una tienda) en una base de datos donde los datos se agregan y limpian para poder plantear consultas de soporte de decisión (por ejemplo, ventas de productos particulares por región). La conversión de datos de los sistemas transaccionales al almacén puede implicar transformaciones y agregación bastante sofisticadas.

1.4.4.1 Esquema de mapas de idiomas.

Formalmente, un mapeo de esquema es un conjunto de expresiones que describen una relación entre un conjunto de esquemas (típicamente dos). En este contexto, las asignaciones de esquema describen una relación entre el esquema mediado y el esquema de las fuentes. Cuando una consulta se formula en términos del esquema mediado, usamos las asignaciones

para reformular la consulta en consultas apropiadas en las fuentes. El resultado de la reformulación es un plan de consulta lógico.

1.4.4.2 Mapeo.

En (Lenzerini 2002) se han propuesto dos enfoques básicos para especificar el mapeo en un sistema de integración de datos, llamado local-as-view (LAV), y global-as-view (GAV), a continuación, se analizará por separado cada enfoque.

Local-as (view LAV)

Desde el punto de vista del modelado, el enfoque LAV se basa en la idea de que el contenido de cada fuente debe caracterizarse en términos de una vista qG sobre el esquema global, donde qG es una consulta sobre el esquema global. Un caso notable de este tipo es cuando el sistema de integración de datos se basa en un modelo empresarial o en una ontología. Esta idea es efectiva siempre que el sistema de integración de datos se base en un esquema global que sea estable y esté bien establecido en la organización. Se tiene en cuenta que el enfoque LAV favorece la extensibilidad del sistema: agregar una nueva fuente simplemente significa enriquecer el mapeo con una nueva aserción, sin otros cambios.

Global-as (view GAV)

Desde el punto de vista de modelado, el enfoque GAV se basa en la idea de que el contenido de cada elemento g del esquema global debe caracterizarse en términos de una vista qS sobre las fuentes. En cierto sentido, el mapeo explícitamente le dice al sistema cómo recuperar los datos cuando uno quiere evaluar los diversos elementos del esquema global. Esta idea es efectiva siempre que el sistema de integración de datos se base en un conjunto de fuentes que sea estable. Tenga en cuenta que, en principio, el enfoque de GAV favorece al sistema al llevar a cabo el proceso de consulta, porque le dice al sistema cómo usar las fuentes para recuperar datos. Sin embargo, ampliar el sistema con una nueva fuente es ahora un problema: la nueva fuente puede de hecho tener un impacto en la definición de varios elementos del esquema global, cuyas vistas asociadas deben ser redefinidas.

Comparación entre LAV y GAV.

Para LAV, el procesamiento de consultas presenta dificultades, puesto que el único conocimiento que se tiene de los datos es una vista en el esquema global que los representa y no se accede realmente a ellos. En contraste, GAV convierte al sistema de integración de datos en un potente procesador de consultas, puesto que existe una recuperación directa de los datos en el esquema de recursos.

LAV permite la incorporación de nuevos recursos con una relativa facilidad, en tanto para GAV esta es una tarea más compleja. Desde el punto de vista de la modelación, mientras que en LAV, el diseñador del sistema ha de concentrar los esfuerzos en la especificación del contenido de los recursos en términos del esquema global, en GAV es necesario especificar concretamente cómo obtener los datos del esquema global a partir de consultas directamente sobre el esquema de recursos.

1.4.5 Integración de datos utilizando ontología.

Uno de los principales desafíos en la tarea de integración es abordar la heterogeneidad de datos. Un enfoque reciente prometedor para abordar este desafío es usar ontologías, modelos conceptuales semánticamente ricos para proporcionar una integración conceptual. Según (Pinkel et al. 2015) la ontología está 'conectada' a las bases de datos con la ayuda de mapeos que son especificaciones declarativas que describen la relación entre el vocabulario ontológico y los elementos del esquema de la base de datos. Las ontologías ya están disponibles en muchos dominios, y muchas de ellas pueden naturalmente ser empleadas para soportar escenarios de integración. Un ejemplo reciente es schema.org una ontología para marcar datos en la web con información de esquema.

1.4.6 Poblado de ontologías.

Señala (Garanina et al. 2017) que el objetivo de poblar una ontología consiste en agregarle algo nuevo, un contenido específico del dominio de un aporte expresado, en particular, en un lenguaje natural. El proceso de la población ontológica es agregar nuevas instancias de conceptos a una ontología. La solución para el poblado de una ontología esta interrelacionada con la elaboración del procesamiento del lenguaje natural (NLP).

También añaden (Blandón Andrade y Zapata Jaramillo 2017) que el proceso de población de ontologías se puede realizar de forma semiautomática o automática. El proceso se debe centrar en enriquecer, por medio de instancias de clases o relaciones, una ontología ya existente, la cual se encuentra vacía. La importancia de poblar ontologías se debe a que las ontologías que existen en los sistemas informáticos se deben actualizar de manera constante y porque, al existir muchas ontologías, se abarcan muchos dominios específicos.

1.4.7 Trabajos relacionados.

Luego de revisar algunas de las fuentes bibliográficas relacionadas con el tema se presenta un resumen de los métodos que proponen los autores de dichas fuentes.

Propone (Pasca 2004) un método para adquirir entidades nombradas en categorías arbitrarias utilizando patrones léxico-sintácticos. También hace referencia al refinamiento de una consulta en una búsqueda web. Las categorías de nombres recogidas se fusionan eficazmente y luego resumen las relaciones semánticas detectadas en los documentos iniciales. Se extraen en pares, por ejemplo: NombreNavegador y Google. Luego, se utilizan los patrones léxico-sintácticos para extraer las instancias. Esos patrones se obtienen de documentos de entrenamiento automáticamente, los cuales constituyen las reglas que se deben cumplir antes de hacer la extracción de las instancias.

Añade (Yoon et al. 2007) un método automático para población de ontologías con datos en formato estructurado. Las instancias se extraen desde páginas web utilizando wrappers o sentencias mediante técnicas de procesamiento de lenguaje natural. El método requiere una ontología e instancias semilla y se extraen instancias desde documentos semiestructurados o no estructurados.

Propone (Ruiz-Martínez et al. 2008) presentan un framework que procesa textos mediante herramientas de procesamiento de lenguaje natural. Realizan las pruebas con la ontología denominada “Travel.owl”, la cual se descarga desde la página de Protégé y a la cual realizan algunos cambios. Utilizan una página web para extraer instancias pertenecientes a la clase “Hotel”.

Según (Danger y Berlanga 2009) lo más aconsejable es una ontología de referencia, reconocedores de entidades y desambiguadores de entidades con el propósito de crear y

combinar adecuadamente un conjunto inicial de instancias. El análisis exhaustivo y la experimentación de la propuesta se llevan a cabo en una variedad de escenarios de aplicación. En el proceso, se define una ontología en formato OWL (Web Ontology Language) que tiene conceptos y relaciones. Existen lexicones que describen las reglas léxicas, para después identificar conceptos y relaciones en el texto. Luego de extraer entidades, se define un conjunto de instancias inicial que usa reglas de inferencia y se genera, finalmente, un conjunto de instancias complejas que definen semánticamente el documento de acuerdo con la ontología dada.

Otro enfoque que le da la literatura al poblado de ontologías, es el de crear nuevas instancias en una ontología determinada partiendo de datos en XML. En este tipo de poblado se concentrará la presente investigación. Según (Cruz y Nicolle 2008) los datos XML se componen de documentos XML y esquemas XML que validan los documentos XML correspondientes. En realidad, un esquema XML contiene el conocimiento que la aplicación debe compartir a través del intercambio de datos. Sin embargo, XML cubre solo el nivel sintáctico, pero no admite el nivel semántico. El nivel semántico describe los significados entre la entrada y la salida. El nivel sintáctico es un conjunto de reglas que permite crear una oración, que dará a la computadora un conjunto de instrucciones para completar una tarea en particular. El nivel léxico se ocupa de las dependencias del dispositivo de entrada en el que el usuario especificará la sintaxis exacta.

Adicionan (Cruz y Nicolle 2008) que los esquemas XML contienen el conocimiento de un dominio que fue especificado por el autor. Esta especificación es solo sintáctica sin ninguna definición semántica. Esto se debe al hecho de que los datos XML se utilizan para intercambiar datos entre procesos que se desarrollaron para estos datos. Para permitir la explotación del conocimiento contenido en esquemas e instancias XML.

1.5 Conclusiones parciales.

Luego de ser consultada y estudiada la literatura especializada, y haberse realizado el marco teórico de la investigación, se concluye lo siguiente:

1- La revisión bibliográfica evidenció que el trabajo con los datos abiertos gubernamentales es un área en constante desarrollo.
2- El estudio de las herramientas para la extracción de datos mencionadas en la bibliografía consultada y el enfoque utilizado por cada una de ellas, evidenció que la herramienta **Apache Tika** posee características superiores a otras similares.
3- Al realizar un estudio de los trabajos relacionados con el poblado de ontologías, el enfoque propuesto por (Cruz y Nicolle 2008) termina siendo el elegido para formar parte de la propuesta de solución en la presente investigación.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

2.1 Introducción.

En este capítulo se presenta un método que integra datos del tipo gubernamental que se encuentran en diferentes formatos, dichos datos terminan siendo integrados a una ontología de dominio (epígrafe 2.2). Como parte de la propuesta ha sido implementada una plataforma informática utilizando herramientas y tecnologías actuales del desarrollo de software (epígrafe 2.3), detallándose los artefactos generados por la metodología de desarrollo adoptada (AUP variación para la UCI). Se aborda acerca del modelo de datos de la propuesta (epígrafe 2.4), los estándares de código empleados (epígrafe 2.5), las técnicas de captura y validación de requisitos, como también el levantamiento de los requisitos funcionales (historias de usuario y sus estimaciones de esfuerzo) y no funcionales (epígrafe 2.6). La arquitectura propuesta del componente (epígrafe 2.7) y la planificación de pruebas (epígrafe 2.8) son apartados tratados de igual modo en este acápite. Se concluye resumiendo los resultados obtenidos durante el diseño e implementación de la propuesta de solución (epígrafe 2.9).

2.2 Descripción de la propuesta general.

Cuando se habla de “Método para la integración de datos heterogéneos mediante una ontología de dominio” se refiere a un conjunto de pasos a tener en cuenta a la hora de agrupar en un mismo sistema datos que se encuentren en documentos con diferentes estructuras y vías de publicación, para luego integrarlos a una ontología de dominio, y ser consultados utilizando las tecnologías de la Web Semántica. Para ello es necesario ver el proceso de poblar una ontología y el proceso de integrar los datos como parte de un solo método, que su entrada serán los datos publicados por la ONEI, estos datos pasarán por diferentes etapas, para luego dar salida a una ontología de dominio cuya información contenida serán los datos antes mencionados, a esto se le llama poblar una ontología.

El método en cuestión consta de 3 etapas, *extracción, integración* y *poblado.* En la primera etapa se utiliza la herramienta **Apache Tika** para extraer los datos de los documentos, esta herramienta es utilizada a nivel de código utilizando sus librerías para manejar los documentos y extraer los datos contenidos en los mismos, los datos extraídos servirán para formar un esquema de alineación en formato XML, se escoge este formato porque simplifica el transporte de la información y los datos se almacenan en texto plano. El esquema de alineación XML es

la entrada al proceso de *integración de los datos,* esta etapa del método tiene como salida los datos integrados listos para poblar la ontología, la implementación de wrappers toma un valor significativo en este paso, debido a que ellos serán los encargados de llevar los datos extraídos de los documentos al esquema de alineación XML previamente definido. La última etapa del método el *poblado de la ontología* tiene como entradas todos los datos integrados en formato XML, el objetivo principal de esta etapa es intanciar las clases de la ontología previamente creada con los datos ya procesados de los documentos de entrada. El método de poblado de ontología que propone (Cruz y Nicolle 2008) es el que se lleva a cabo en la presente tesis debido a su semejanza con la misma.

Según (Cruz y Nicolle 2008) el método se articula en dos pasos. El primer paso se relaciona con la formalización de las reglas de "alineación" entre un esquema XML y un esquema OWL XML. Las reglas de " alineación" permiten enriquecer una ontología de dominio existente a partir de conceptos y relaciones conceptualmente presentes en los esquemas. Este paso también se relaciona con la definición de reglas de " alineación básica" para traducir datos complejos como árboles secundarios en atributos simples o complejos de una clase. La segunda etapa consiste en poblar la ontología enriquecida previamente a partir de documentos XML validados por el esquema XML alineado (Figura 2 (P)).

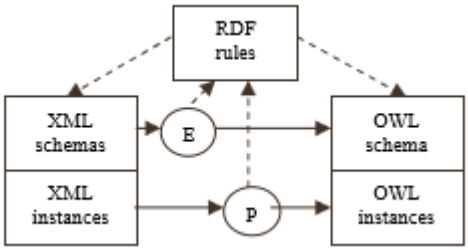

Figura 2. *Enriquecimiento y población de una ontología existente. (Fuente: Ontology Enrichment and Automatic Population From XML Data).*

La población tiene que seguir algunas reglas, como la imitación de atributos cardinales e instancias únicas. En consecuencia, las reglas de "alineación básica" deben modelar y especificar restricciones en los atributos. Para especificar los elementos relevantes de un esquema XML para el proceso de enriquecimiento, es necesario identificar y marcar estos elementos. Estas marcas se llaman "marcas esquemáticas" y son anotaciones RDF externas de estructuras XML.

En realidad, las "marcas esquemáticas" son específicas de un gráfico RDF que define anotaciones externas en esquemas XML. En la Figura 2, (E) es el proceso de enriquecimiento

de una ontología y (P) es el proceso de poblar la ontología. Estos dos procesos usan un gráfico RDF como reglas para enriquecer y poblar la ontología. Las reglas en RDF se definen durante el proceso de mapeo.

El método se basa en la definición de marcas esquemáticas, reglas de mapeo y reglas avanzadas de mapeo. La primera parte describe el marcado del esquema para anotar el elemento de los esquemas XML. Se relaciona con el esquema XML, pero también con el esquema OWL XML. La segunda parte presenta el paso de mapeo que está compuesto por las reglas de conversión, el proceso de enriquecimiento ontológico y el proceso de población ontológica.

Marcas esquemáticas

Adiciona (Cruz y Nicolle 2008) que un gráfico RDF se usa para anotar cada esquema XML. Estas marcas se especifican para mantener toda la información en un gráfico que se requiere durante el mapeo con el paso de la marca esquemática OWL. La sintaxis TriG se utiliza para facilitar las explicaciones de cómo emplear las marcas esquemáticas con los esquemas XML. El ejemplo 1 muestra cómo marcar un primer esquema y cómo especificar un id. Para evitar instancias duplicadas de OWL. El espacio de nombre "bmr" se usa para identificar las reglas de mapeo básicas. Además, "bmr: xpath" define un elemento en el esquema XML como lo hace "bmr: xpathId" con la adición de que este último actúa como un identificador. Una propiedad única en un elemento XML es bienvenida para definir un identificador.

```
# TriG RDF mapping
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
 @prefix owl: <http://www.w3.org/2002/07/owl#>.
 @prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
 @prefix bmr: <http://www.example.org/BasicMappingRules#>.
 @prefix amr: <http://www.example.org/AdvancedMappingRules#>.: XSDMarksSchema_1 {:
student bmr:xpath "/univ/students/student" .
  : student Name bmr:xpath "/univ/students/student/name" .
  : prof bmr:xpath "/univ/profs/prof" .
  : profName   bmr:xpath "/univ/profs/prof" .
  : profAge bmr:xpath "/univ/profs/prof/age" .
  : studentAge bmr:xpath "/univ/ students/student/age" . …
  : studentId bmr:xpathId      "/univ/students/student[@idNum]" .
  : prof Id bmr:xpathId      "/univ/profs/prof[@idprof]" .
```

Ejemplo 1 Definiciones de marcas esquemáticas en un esquema XML (Fuente: Ontology Enrichment and Automatic Population From XML Data)

Reglas de conversión

Afirma (Cruz y Nicolle 2008) que las reglas básicas de mapeo definen reglas para la conversión de datos de esquemas XML a OWL. Las reglas de mapeo avanzado usan reglas de mapeo básicas para definir el mapeo entre esquemas XML y OWL. Además, estas reglas permiten definir nuevos elementos en la ontología OWL para el proceso de enriquecimiento y para el proceso de población.

Población ontológica

Agrega (Cruz y Nicolle 2008) que para generar la población de la ontología, se debe definir el mapeo RDF. Para lograr esto, se deben crear relaciones entre el gráfico RDF de las marcas de esquema XML, el gráfico RDF de las marcas esquemáticas OWL y el gráfico RDF de las reglas de conversión.

Cuando el proceso de poblado de la ontología de dominio se realice se podrá dar cumplimiento a los siguientes objetivos:

- Resolver un problema de integración de datos al no contar en la Universidad de Ciencias Informáticas con un proceso para dicho propósito.
- Ayudar en la toma de decisiones al poder acceder de forma directa a los datos. Esto permite mayor claridad a la hora de la consulta.

2.3 Metodología, herramientas y técnicas.

2.3.1 Metodología de desarrollo.

El desarrollo de todo software debe estar guiado por una metodología de desarrollo. De esta depende, en gran medida, que el software tenga la calidad requerida. Existen dos grupos de metodologías: ágiles y tradicionales. No existe una metodología universal para cada tipo de proyecto. Se define una metodología según las características del equipo de desarrollo, el dominio de aplicación, el tipo de contrato, la complejidad y la envergadura del proyecto.

Dada la necesidad de desarrollar la propuesta de solución en un breve período de tiempo, garantizando además la flexibilidad necesaria en cuanto a la variación de los requisitos y el manejo de los riesgos técnicos, así como reducir la generación de documentos y artefactos, y

no habiendo un contrato tradicional, siendo el cliente parte del equipo de desarrollo; se hace necesario optar por un enfoque ágil de desarrollo de software en lugar de un enfoque tradicional o pesado. Teniendo en cuenta lo anterior han sido evaluadas varias metodologías que siguen el enfoque ágil de desarrollo de software, tal es caso de Scrum (Sutherland 2015), Agile Unified Process (AUP, por sus siglas en inglés) (Edeki 2013) , Extreme Programming[11] (XP, por sus siglas en inglés) (Kniberg 2015) y AUP-UCI.

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP), en unión con el modelo CMMI-DEV v1.3 la cual es una versión simplificada del Proceso Unificado de Desarrollo (RUP por sus siglas en inglés) (Sánchez 2014).

Esta metodología es una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP tiene entre sus principales características que está dirigida por pruebas, cuenta con un modelado ágil, incluye la gestión de cambios ágiles y la refactorización de base de datos para mejorar la productividad (Sánchez 2014).

Variación AUP para la UCI

Adiciona (Sánchez 2014) que al no existir una metodología de desarrollo de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide utilizar una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Estará apoyada en el Modelo CMMI-DEV v1.3, el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad (Sánchez 2014).

Escenarios para la disciplina de requisitos:

La metodología propone cuatro escenarios para la disciplina de requisitos. El presente trabajo estará regido por el escenario número 4 y su selección está basada en que entre sus

[11] http://www.extremeprogramming.org, www.xprogramming.com

características cumple que: se ajusta para proyectos que no modelan un negocio sino modelan el sistema con las Historias de Usuario (HU, por sus siglas en español), siendo éste el caso que ocupa. No es un proyecto extenso y el cliente estará siempre acompañando al equipo de desarrollo para definir en conjunto los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

2.3.2 Entorno de trabajo.

Un Integrated Development Environment (IDE, por sus siglas en inglés), en español: Entorno de Desarrollo Integrado, es un programa compuesto por una serie de herramientas utilizadas por programadores para desarrollar aplicaciones (Mahmood y Reddy 2014). Para el desarrollo de la solución, como Entorno de Desarrollo Integrado (IDE según sus siglas en inglés) se seleccionó el **NetBeans**[12] (v.8.1). Es libre, de código abierto y contiene todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es mundialmente conocido por su integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, permite la utilización y edición de los componentes visuales de una forma sencilla. Es un entorno de desarrollo, disponible para varios sistemas operativos como Windows, Linux, Mac y Solaris. Además, incluye el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente.

2.3.3 Lenguaje de programación.

Java[13] es un lenguaje de programación creado por *Sun Microsystems Inc* en un proceso por etapas que arranca en 1990 basado en la programación orientada a objetos. Una de las características más importantes de Java es que los programas ejecutables, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. Java aprovecha características de la mayoría de los lenguajes modernos evitando sus inconvenientes. Tiene una gran funcionalidad gracias a sus librerías (clases). El manejo de la memoria no es un problema, la gestiona el propio lenguaje y no el programador. Incorpora

[12] http:// netbeans.org/
[13] https:// www.java.com/es/about/

Multi-Threading para permitir la ejecución de tareas concurrentes dentro de un mismo programa.

2.3.4 Herramientas.

Apache Jena [14] es un framework de Web Semántica de código abierto para Java. Proporciona una API para extraer datos y escribir en gráficos RDF. Los gráficos se representan como un "modelo" abstracto. Un modelo se puede obtener con datos de archivos, bases de datos, URL o una combinación de estos. Un modelo también se puede consultar a través de SPARQL. Jena proporciona soporte para OWL (Web Ontology Language). El framework tiene varios razonadores internos y el razonador Pellet (un razonador Java OWL-DL de código abierto) puede configurarse para funcionar en Jena.

Maven [15] es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl), pero tiene un modelo de configuración de construcción más simple, basado en un formato XML. Estuvo integrado inicialmente dentro del proyecto Jakarta pero ahora ya es un proyecto de nivel superior de la Apache Software Foundation. Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

2.4 Estándares de código.

Un estándar de código se basa en la estructura y apariencia física de un programa con el fin de facilitar la lectura, comprensión, mantenimiento del código, reutilización a lo largo del proceso de desarrollo de un software y no en la lógica del programa. Un estándar de programación no solo busca definir la nomenclatura de las variables, objetos, métodos y funciones, sino que también tiene que ver con el orden y legibilidad del código escrito (Guerrouj 2013). Partiendo de lo dicho anteriormente, se definen tres partes principales dentro de un estándar de programación:

[14] https://jena.apache.org/
[15] https://maven.apache.org/

2.4.1 Nomenclatura de las clases.

Los nombres de las clases siempre comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación UpperCamelCase, la cual define que la primera letra de cada una de las palabras es mayúscula y con solo leerlo se reconoce el propósito de la misma.

Ejemplo: ExtraccionIntegracion. En este caso el nombre de la clase está compuesto por dos palabras iniciadas cada una con letra mayúscula.

2.4.2 Nomenclatura de las funcionalidades y atributos.

El nombre a emplear para las funciones y los atributos se escriben con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCase.

Ejemplo: WriteFile (). El nombre de este método está compuesto por dos palabras, por lo tanto, se utiliza la notación antes mencionada.

2.4.3 Nomenclatura de los comentarios.

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar para lograr una mejor comprensión del código.

2.5 Requisitos.

La definición de requisito en la literatura científica cuenta con varias acepciones. La Real Academia Española[16] lo define como: circunstancia o condición necesaria para algo. Por otra parte, IEEE en (ISO 2011) enuncia el concepto de la siguiente manera: declaración que se traduce o expresa una necesidad y sus limitaciones y las condiciones correspondientes. A continuación de listan las técnicas empleadas para la captura de requisitos, los requisitos funcionales y no funcionales identificados y las técnicas para su validación.

2.5.1 Técnicas de captura de requisitos.

Las técnicas de captura de requisitos trabajadas en la presente investigación son:

[16] http://dle.rae.es/?id=W6xh4wt

Tormenta de ideas

Las tormentas de ideas (del inglés: brainstorming) son una técnica de reuniones en grupo sencilla y fácil de aplicar, cuyo objetivo es que los participantes muestren sus ideas en un ambiente libre de críticas o juicios. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. Las tormentas de ideas suelen ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.

2.5.2 Requisitos funcionales.

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactarlos. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de como este se debe comportar en situaciones particulares.

La siguiente tabla muestra el listado de los requisitos funcionales identificados:

***Tabla 2**.Requisitos funcionales del software. (Fuente: elaboración propia).*

No.	**Requisito funcional**	**Prioridad para el cliente**	**Complejidad**
1	Extraer datos de los documentos.	Alta	Alta
2	Integrar los datos.	Alta	Alta
3	Poblar ontología.	Alta	Alta

2.5.3 Historias de Usuario.

Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles puesto que son una forma rápida para la especificación y administración de requisitos, sin necesidad de elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes y deben cumplir varias restricciones, entre ellas: ser independientes unas de otras, negociables, estimables, pequeñas y verificables, por mencionar algunas.

Tabla 3.*Historia de usuario del requisito: Extraer datos de los documentos***.**

Código: HU-1	**Nombre del requisito:** Extraer datos de los documentos.
Programador: Dasiel Miguel Pedrero Mesa	*Iteración asignada*: 1
Prioridad: alta	*Tiempo estimado*: 160 horas
Riesgo en desarrollo: alto	*Tiempo real*: 80 horas
Descripción: El propósito de la actividad es extraer los datos de los documentos proporcionados por la ONEI.	

Tabla 4*.Historia de usuario del requisito: Integrar los datos*

Código: HU-2	**Nombre del requisito:** Integrar los datos.
Programador: Dasiel Miguel Pedrero Mesa	*Iteración asignada*: 1
Prioridad: alta	*Tiempo estimado*: 176 horas
Riesgo en desarrollo: alto	*Tiempo real*: 120 horas
Descripción: El propósito de la actividad es a través de uno de los métodos de integración de datos existentes, llevar los datos a un mismo formato para un correcto uso de los mismos.	

Tabla 5*.Historia de usuario del requisito: Poblar ontología.*

Código: HU-3	**Nombre del requisito:** Poblar ontología.
Programador: Dasiel Miguel Pedrero Mesa	*Iteración asignada*: 1
Prioridad: alta	*Tiempo estimado*: 176 horas
Riesgo en desarrollo: alto	*Tiempo real:* 120 horas
Descripción: La actividad en cuestión se considera la más importante dentro del sistema, ya que de este proceso saldrán los datos incorporados a una ontología de dominio, siendo esto la salida del método de integración de datos.	

Estimación de esfuerzo por historias de usuario.

Dependiendo de la prioridad asignada por el cliente a cada historia de usuario y atendiendo a la complejidad y riesgo determinado por el programador, se define la estimación de cada una de las historias de usuario identificadas. La Tabla 6 muestra los resultados de la estimación realizada. La unidad de estimación es el punto, y un punto equivale a las horas de programación.

***Tabla 6**.Estimación de esfuerzo por historia de usuario.*

Historias de usuario	**Puntos de estimación**
Obtener los datos de las fuentes	*80*
Integrar los datos	*120*
Poblar ontología	*120*

2.5.4 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 (Moreno y Marciszack 2013). Ver Tabla 7.

***Tabla 7**.Relación de los requisitos no funcionales identificados*

Software	**Hardware**

La PC donde se despliegue la aplicación debe tener las siguientes propiedades: Protégé v (5.2.0) Máquina Virtual de Java v (1.7).	La PC donde se despliegue la aplicación debe tener las siguientes propiedades: Microprocesador: Intel Core i3 de cuarta generación a 2.33 GHz. Memoria RAM: 4 GB DDR3 SDRAM. Capacidad de almacenamiento: 500 GigaBytes. Tarjeta de red: Fast Ethernet a 100 Mbps (megabits por segundo).

2.5.5 Técnicas de validación de requisitos.

Como técnicas de validación de requisitos adoptadas en este trabajo se tiene:

Casos de prueba

Los casos de prueba son un conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación o sistema software es parcial o completamente satisfactoria. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba. Lo que caracteriza un escrito formal de caso de prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba. La entrada conocida debe probar una precondición y la salida esperada debe probar una postcondición.

2.6 Diagrama de clases.

El diagrama de clases muestra como las clases implementadas dependen una de la otra y todas tributan a la llamada de la clase principal. Ver Figura 3.

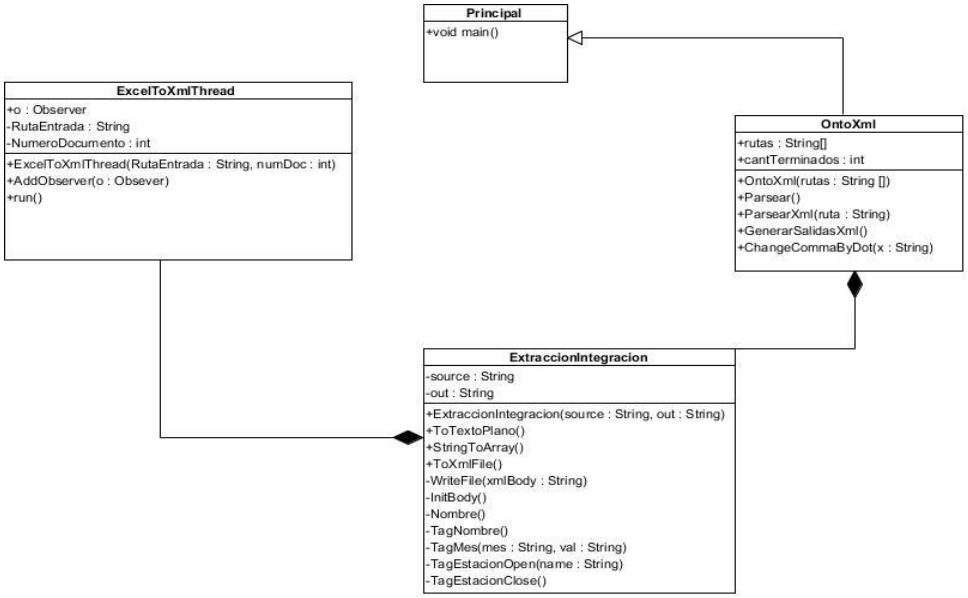

Figura 3. *Diagrama de clases. (Fuente: elaboración propia).*

2.1 Arquitectura.

El método de integración de datos como propuesta de solución sigue un estilo arquitectónico de flujo de datos, que enfatiza la reutilización y modificalidad; siendo apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos. Se recomienda ser aplicada en escenarios donde se demande el uso de un índice para lograr mejores tiempos de respuesta a consultas formuladas por usuarios para el acceso a un recurso. La arquitectura utilizada es de tuberías y filtros, que consiste en ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales, siendo la entrada de cada una la salida de la anterior. Una tubería (del inglés, pipeline) conecta componentes computacionales (filtros) a través de conectores (del inglés, pipes), de modo que las computaciones se ejecutan a la manera de un flujo. Los datos se transportan a través de las tuberías entre los filtros, transformando gradualmente las entradas en salidas. En la Figura 4 se muestra la propuesta del método de integración de datos heterogéneos a una ontología de dominio siguiendo la arquitectura anteriormente descrita.

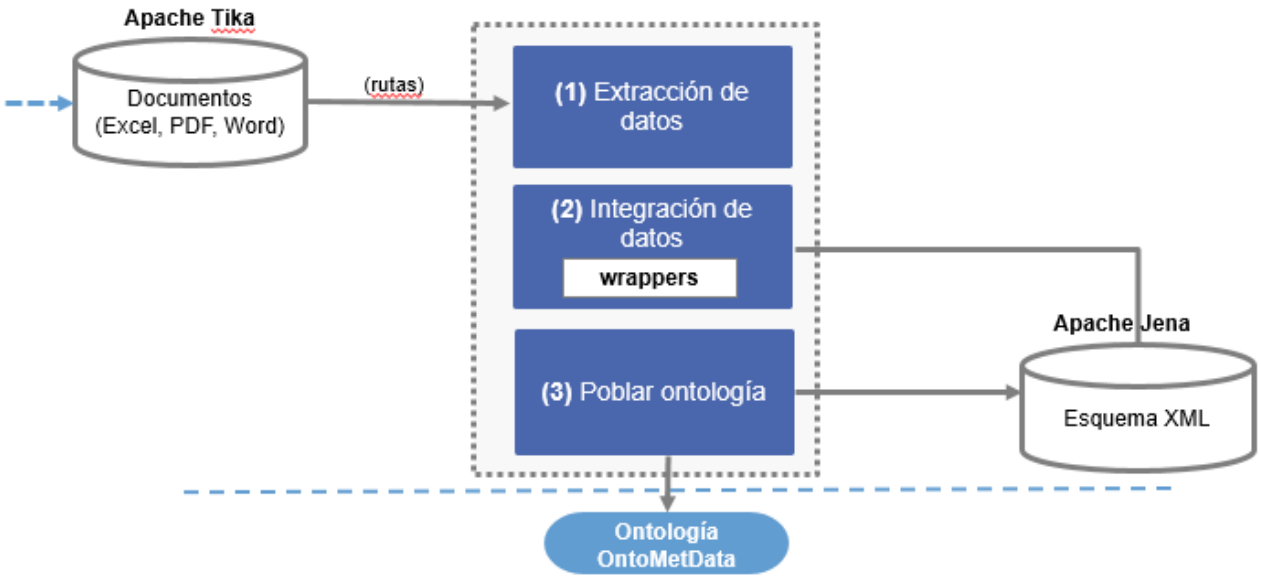

Figura 4. *Arquitectura del método de integración de datos a una ontología de dominio. (Fuente: elaboración propia).*

2.7 Modelo de datos.

El modelo de datos de la propuesta de solución se compone de entidades, propiedades y valores representados en una ontología. La ontología propuesta está diseñada para almacenar datos relacionados con los valores meteorológicos emitidos por las diferentes estaciones de todas las provincias de Cuba. La misma está representada e implementada con el único objetivo de ser poblada con los datos gubernamentales emitidos por las estaciones anteriormente mencionadas como se muestra en la Figura 5.

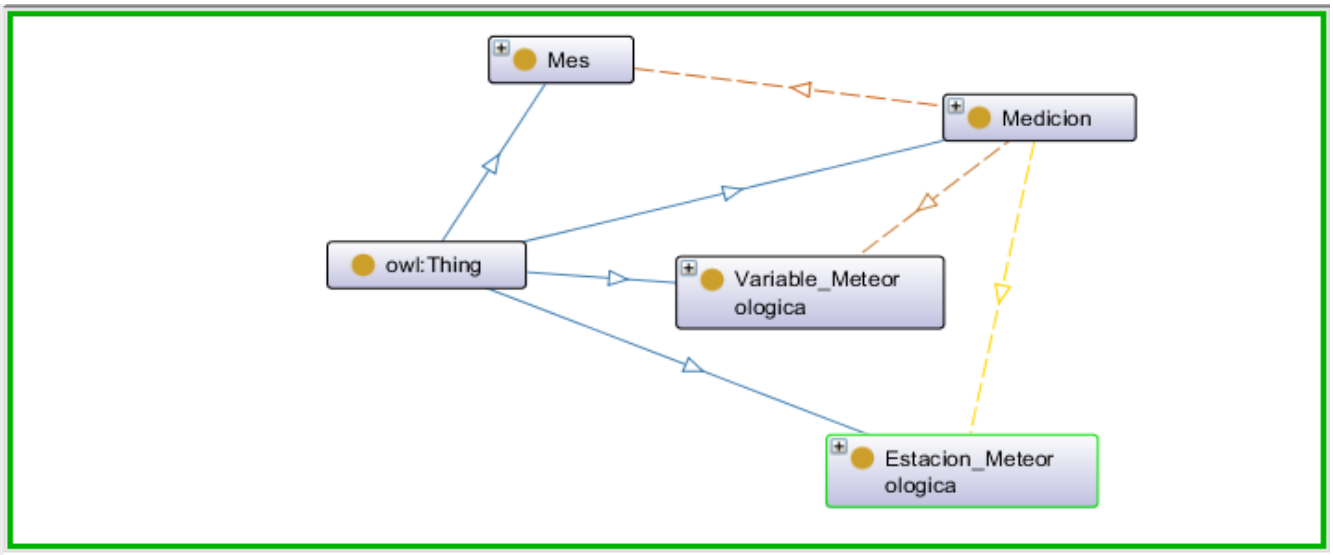

Figura 5. *Modelo de datos del proyecto "Ontología OntoMetData". (Fuente: elaboración propia).*

2.8 Patrones de diseño.

Un patrón es una solución recurrente a un problema dentro de un contexto dado. Los patrones surgen de la experiencia de seres humanos al tratar de lograr ciertos objetivos, además capturan la experiencia existente y probada para promover buenas prácticas.

Para el caso específico de la propuesta de solución se emplea un grupo de patrones relacionados con el diseño de software, llamados patrones GRASP (siglas en inglés de Patrones Generales de Software para Asignación de Responsabilidades), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

A continuación, se relaciona el uso de los patrones de diseño utilizados a través de ejemplos:

Patrón Experto:

El objetivo de éste patrón es asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón se ve reflejado en la clase OntoXml.java la cual espera porque la clase ExtraccionIntegracion.java termine sus funcionalidades para ella comenzar con las suyas.

Patrón Creador:

Este patrón guía la asignación de responsabilidades relacionadas con la creación de instancias, tarea muy frecuente en los sistemas orientados a objetos. Para la propuesta de solución, este patrón se utiliza en la mayoría de las clases de la aplicación.

Patrón Alta Cohesión:

Con este patrón se espera que una clase tenga un número moderado de responsabilidades dentro de un área funcional y colabore con las otras para llevar a cabo una tarea. En el caso de la propuesta de solución utilizando el patrón experto se le asigna a cada clase las responsabilidades que le corresponden y se establecen las condiciones para que una clase colabore con las demás en la resolución de tareas que las implican a todas y que no son capaces de resolver por sí solas.

Patrón Bajo Acoplamiento:

Al programar o diseñar se debe lograr un acoplamiento lo más bajo posible entre dos unidades de software cuales quieran que estas sean. Esto redunda en una mejora considerable en la detección y corrección de errores, en una mayor facilidad de mantenimiento y, sobre todo, en la reutilización.

En el caso de la propuesta de solución se logra un bajo nivel de acoplamiento entre la extensión implementada y la herramienta Apache Tika. De esta forma, si la extensión es retirada (incluso en tiempo de ejecución), no se altera el funcionamiento de Apache Tika.

2.9 Conclusiones parciales.

En este capítulo se propuso un método para la integración de datos heterogéneos gubernamentales a una ontología de dominio. Luego de presentada la propuesta, se concluye que:

1- El diseño de la arquitectura del componente, 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 método desarrollado.
2- El método de integración de datos heterogéneos gubernamentales consta de tres etapas bien definidas. Se implementa un prototipo funcional con el fin de cargar las rutas de los documentos del tipo gubernamental y mostrar una ontología de dominio con los datos de los documentos previamente procesados
3- El método desarrollado cumple con todas las especificaciones de los requisitos seleccionados y se ajusta a la arquitectura de software propuesta.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

3.1 Introducción.

En el capítulo anterior fueron definidos los tipos y técnicas de pruebas para su empleo posterior. Este capítulo, tiene como objetivo validar la propuesta de solución (epígrafe 3.3) aplicando las pruebas descritas (epígrafe 3.2). Se desarrolla un caso de estudio empleando datos reales provenientes de revistas cubanas (epígrafe 3.4). Se describen en detalle los principales resultados obtenidos con el caso de estudio y el experimento desarrollado (epígrafe 3.5). Por último, se relacionan aspectos representativos resultantes de la validación de la propuesta de solución (epígrafe 3.6) y se emiten las consideraciones generales de esta sección (epígrafe 3.7).

3.2 Planificación de pruebas.

El proceso de pruebas de AUP constituye una de sus fortalezas, puesto que permite aumentar la calidad del sistema reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. En la variación de la metodología AUP para la UCI este proceso se desagrega en tres disciplinas: (1) pruebas internas, (2) pruebas de liberación y por último (3) pruebas de aceptación. En el caso específico del componente propuesto, solo se tratan las pruebas internas y las pruebas de aceptación, debido a que las pruebas de liberación son diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

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. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. El objetivo de las pruebas internas es el aislamiento de partes del código y la demostración de que estas partes no contienen errores.

Las pruebas de caja blanca, denominada a veces “prueba de caja de cristal”, a consideración de (Pressman 2005), es un método de diseño de casos de prueba que se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar.

Este tipo de prueba será aplicada a la estructura de control del diseño procedimental (código fuente), a través de la técnica del camino básico. La prueba del camino básico es una técnica de prueba de caja blanca propuesta por Tom McCabe, como expone (Pressman 2005). Esta técnica, permite obtener una medida de la complejidad lógica de un diseño y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Mediante su aplicación: se garantiza la ejecución por lo menos una vez de los caminos independientes de cada módulo, se ejercitan las decisiones lógicas y estructuras internas de datos para asegurar su validez, así como ejecución de los bucles.

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

3.2.2 Pruebas de aceptación.

Estas pruebas las realiza el cliente. En este caso el cliente es el Grupo de Investigación de Web Semántica. Las pruebas de aceptación, no se realizan durante el desarrollo, pues no se podrían presentar al cliente; sino que se ejecutan sobre el producto terminado e integrado o bien una versión del producto o una iteración funcionad pactada previamente con el cliente. Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está

listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que aparecen cuando el cliente comienza a usarlo, por lo que las pruebas de aceptación tienen un mayor grado de importancia que las pruebas unitarias (Heredia, Álvarezza y Linares 2011).

Las pruebas de aceptación, son pruebas funcionales, sobre el sistema, y buscan una cobertura de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a probar cuando una historia de usuario ha sido correctamente implementada. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas (Pressman 2005).

Las pruebas de caja negra, también denominada “prueba de comportamiento”, se centran en los requisitos funcionales del software (Pressman 2005). Mediante las técnicas de prueba de caja negra se obtiene un conjunto de casos de prueba que intentan encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación.

3.3 Prueba de software.

El principal objetivo del diseño de casos de prueba es obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Para llevar a cabo este objetivo, se usan dos categorías diferentes de técnicas de diseño de casos de prueba: prueba de caja blanca y prueba de caja negra.

3.3.1 Prueba de caja blanca.

El uso de esta técnica es mostrado en el ejemplo siguiente. Se analizan y enumeran las sentencias de código del método ToXmlFile() contenidas en la clase ExtraccionIntegracion.java. Este método construye y escribe un archivo XML proveniente de un documento en formato Excel. Ver la Figura 6.

```
public void ToXmlFile() throws IOException, SAXException, TikaException {
    String XmlBody = "";                                                    //1
    XmlBody += initBody();                                                  //1
    XmlBody += tagNombre();                                                 //1
    String Arreglo[][] = stringToArray(toTextoPlano());                     //1

    for (int i = 1; i < Arreglo.length; i++) {                              //2
        XmlBody += tagEstacionOpen(Arreglo[i][0]);                          //3

        for (int j = 1; j < 13; j++) {                                      //4
            XmlBody += tagMes(Arreglo[0][j], Arreglo[i][j]);                //5
        }
        XmlBody += tagEstacionClose();                                      //6
    }

    XmlBody += "</documento>";                                              //7
    System.out.println(XmlBody);                                            //7
    WriteFile(XmlBody);                                                     //7

}                                                                           //8
```

Figura 6.*Código del método ToXmlFile(). (Fuente: elaboración propia).*

A continuación, se muestra el grafo de flujo correspondiente al código del método seleccionado:

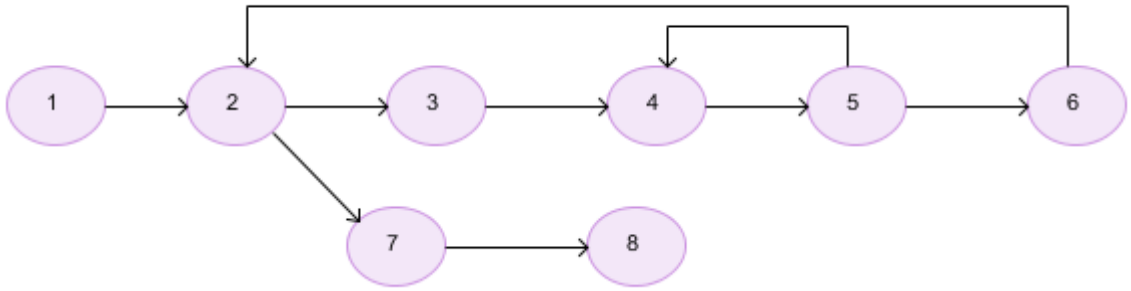

Figura 7.*Grafo de flujo. (Fuente: elaboración propia).*

Complejidad ciclomática:

Existen varias formas de calcular la complejidad ciclomática de un programa a partir de un grafo de flujo:

1. V(G) = A – N + 2

A: número de aristas, N: número de nodos.

V(G)= (9 - 8) + 2 = 3

2. V(G) = NP + 1

NP: número de nodos predicados (nodos de los cuales parten dos o más aristas). V(G)= 2 + 1 = 3

3. V(G)= R

R: número de regiones del grafo. V(G)= 3

Luego de calculada la complejidad ciclomática mediante las tres fórmulas se evidencia que la complejidad es 3, lo cual indica que existen 3 caminos independientes, representando este valor el mínimo número de casos de pruebas.

Los caminos independientes resultantes son:

Camino 1: 1-2-7-8

Camino 2: 1-2-3-4-5-6-2-7-8

Camino 3: 1-2-3-4-5-4-5-6-2-7-8

Se procede a ejecutar los casos de pruebas para cada uno de los caminos básicos determinados en el grafo de flujo. Para definir los casos de prueba es necesario tener en cuenta: descripción, condición de ejecución, entrada y resultados esperados. Ver Tabla 8.

Tabla 8. *Caso de prueba del camino básico No. 1*

Código: CPCB-01	
Descripción:	Este procedimiento genera y escribe un documento XML proveniente de los documentos en formato Excel.
Condición de ejecución:	La cantidad de columnas del documento deberá ser igual a la cantidad de columnas del arreglo.
Entrada:	Documentos Excel llevados a texto plano.
Resultados esperados:	Documento XML.
Evaluación de la prueba: Satisfactoria	

Los otros 2 casos de prueba se encuentran en el anexo 1 de la presente investigación.

Análisis de resultados:

En una primera iteración se realizaron un total de 3 casos de pruebas de caja blanca, de los cuales resultaron satisfactorios 2, lo que representa el 66% del total de los casos de prueba. Ver Figura 8.

Figura 8. *Resultados de las pruebas de caja blanca. (Fuente: elaboración propia).*

Una vez corregidos los errores detectados en la primera iteración de las pruebas se realiza una segunda iteración con un total de 3 casos de pruebas resultando el 100% de ellos satisfactorios y siendo corregidos los errores detectados tras la primera iteración.

3.3.2 Pruebas de caja negra.

Para la realización de las pruebas de caja negra se empleó la técnica partición de equivalencia. Esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software. A continuación, es sometida a pruebas de aceptación las historias de usuario definidas en el capítulo anterior.

Tabla 9.*Caso de prueba de aceptación CP-01.*

Código: CP-01		**Historia de Usuario:** HU-1
Nombre: Extraer los datos de los documentos.		
Descripción: En este caso de prueba se verifica el procedimiento que se realiza cuando se toman varios documentos para extraerle los datos.		
Acción a probar:	*Datos de entrada:*	*Resultados esperados:*

Introducir rutas de los documentos a procesar.	Documentos Excel.	Los datos deben ser extraídos del documento inicial.
Mostrar los datos de los documentos en texto plano.	Documentos Excel.	Los datos de los documentos se mostrarán en la consola.
Evaluación de la prueba:	Satisfactoria	

Tabla 10. *Caso de prueba de aceptación CP-02.*

Código: CP-02		**Historia de Usuario:** HU-2
Nombre: Integrar los datos.		
Descripción: En este caso de prueba se verifica el procedimiento que se realiza cuando se toman los datos extraídos de los documentos y se procede a integrarlos en un esquema XML.		
Acción a probar:	*Datos de entrada:*	*Resultados esperados:*
Almacenar los datos en un arreglo bidimensional.	Datos en texto plano.	Los datos de los documentos deben ser integrados en un documento en formato XML.
Mostrar los datos de los documentos en formato XML.	Datos en texto plano.	Los datos de los documentos en formato XML se mostrarán en la consola.
Evaluación de la prueba:	Satisfactoria	

Tabla 11. *Caso de prueba de aceptación CP-03.*

Código: CP-03	**Historia de Usuario:** HU-3

Nombre: Poblar ontología.		
Descripción: En este caso de prueba se verifica el procedimiento que se realiza cuando se toman los datos contenidos en un esquema XML y se introducen dentro de la ontología.		
Acción a probar:	*Datos de entrada:*	*Resultados esperados:*
Parsear XML	Datos en un esquema XML.	Los datos contenidos en el esquema XML serán parseados y se podrán acceder a ellos.
Instanciar las clases de la ontología con los datos del esquema XML.	Datos en un esquema XML.	Las clases de la ontología serán instanciadas a la vez que el esquema XML es recorrido, completando la fase de poblado de la ontología.
Evaluación de la prueba:	Satisfactoria	

Análisis de resultados:

Se realizaron un total de 12 casos de pruebas de caja negra, de ellos 2 resultaron no satisfactorios, lo cual representa el 17% del total de casos de prueba de caja negra realizados, mientras los 10 casos de prueba restantes resultaron satisfactorios para un 83% del total. Ver Figura 9.

Figura 9. *Resultados de las pruebas de caja negra. (Fuente: elaboración propia).*

Durante la realización de las pruebas de caja negra, en los 2 casos de prueba que resultaron no satisfactorios fueron detectados 5 errores, de ellos: 4 funciones incorrectas y 1 error de rendimiento. Cada error encontrado durante las pruebas realizadas fue mitigado.

3.4 Caso de estudio.

Con el objetivo de validar la solución al problema de investigación se diseña un caso de estudio. Se utiliza para ello una colección con 30 Excel los cuales están almacenados a priori en un directorio local para ser procesados desde el mismo. Los documentos en este caso a analizar son los registros de temperatura y lluvia emitidos por todas las estaciones meteorológicas de Cuba. Para el caso de estudio se cuenta con un equipo de cómputo con las siguientes prestaciones:

- Tipo de CPU: Intel Core i3 5200U (5ta generación) a 2.33GHz
- Memoria del sistema: 4GB RAM DDR3 SDRAM

Se proponen los siguientes escenarios para la evaluación:

1. Realizar la extracción e integración de datos gubernamentales a una ontología sin el empleo de la propuesta de solución, es decir de manera manual.

2. Realizar la extracción e integración de datos gubernamentales a una ontología utilizando la propuesta de solución como estímulo.

Realizar el método de extracción e integración de datos gubernamentales de forma manual implica la acción de abrir los documentos, copiar todos los valores que contengan, introducirlos en la ontología utilizando el Protégé. Para ello es necesario tener la ontología previamente cargada, acceder a las clases de la misma, crear las instancias y asociarlas con las propiedades. Todo esto se realiza tantas veces como documentos se necesiten procesar.

3.5 Diseño experimental.

De acuerdo a la clasificación de los experimentos mostrada al inicio del capítulo, se emplea en la investigación un pre-experimento, dado que se precisa el resultado de una observación inicial que será comparada en otro momento con la aplicación de un estímulo. Se definen cinco tareas a realizar, enumeradas seguidamente:

1. Procesar 5 documentos en formato Excel.

2. Procesar 10 documentos en formato Excel.

3. Procesar 15 documentos en formato Excel.

4. Procesar 20 documentos en formato Excel.

5. Procesar 30 documentos en formato Excel.

A continuación, se muestra la Tabla 12 con el diseño experimental propuesto.

Tabla 12. *Diseño experimental propuesto. (Fuente: elaboración propia).*

Fuente de datos	Tareas	Observación simple	Estímulo	Observación con estímulo
G	T1	OS1	E	OE1
G	T2	OS2	E	OE2
G	T3	OS3	E	OE3
G	T4	OS4	E	OE4
G	T5	OS5	E	OE5

La simbología empleada en la tabla anterior es la siguiente:

- G: Colección de documentos en formato Excel como fuente de datos.

- Ti: Tareas (procesar X cantidad de Excel) realizadas sobre G. El subíndice i representa el número de la consulta.

- OSi: Resultado de la observación luego de acometer T. El indicador es el tiempo en segundos que tarda el usuario en extraer e integrar los datos datos manualmente.

- E: Tratamiento o estímulo. En este caso la aplicación del método propuesto.

- OEi: Observación realizada tras aplicar E. El indicador es el tiempo en segundos que tarda el método en realizar la extracción e integración.

3.6 Análisis de resultados.

Tabla 13. *Análisis de resultados del experimento*

Fuente de datos	Tareas	Observación simple	Estímulo	Observación con estímulo
G	T1	OS1: 00:15:03:33	E	OE1: 00:00:07:90
G	T2	OS2: 00:29:34:00	E	OE2: 00:00:08:71
G	T3	OS3 : 01:05:25:46	E	OE3: 00:00:10:07
G	T4	OS4: 01:57:53:59	E	OE4: 00:00:11:18
G	T5	OS5: 02:00:03:06	E	OE5: 00:00:11:62

La Tabla 13 corresponde a la aplicación de la propuesta de solución como estímulo en el segundo escenario de prueba. Nótese que los tiempos de procesamiento de manera manual, en este caso la observación simple, son muy superiores a los tiempos obtenidos al aplicar el estímulo en este caso el método de extracción e integración de los datos al mismo grupo de tareas. Es decir, que al aplicar el estímulo se puede apreciar una reducción considerable de los tiempos que toma la extracción, integración y poblado de los datos desde los documentos en formato Excel, quedando validada de esta forma la propuesta de solución.

3.7 Conclusiones parciales.

En este capítulo se diseñó un caso de estudio y se propuso un diseño experimental con el propósito de validar la propuesta de solución presentada en el capítulo anterior. Tras aplicar las pruebas de software definidas y el diseño experimental descrito, se concluye lo siguiente:

1- El caso de estudio permitió evaluar el método propuesto a partir de los resultados obtenidos mediante su ejecución en un prototipo funcional implementado.
2- La validación de la solución propuesta estuvo conducida por la realización de experimentos, demostrando la validez de los resultados obtenidos.
3- El análisis del experimento aplicado demostró que la aplicación de un método de integración de datos mediante una ontología reduce el tiempo que demoraría si se realizara el procedimiento de forma manual.

CONCLUSIONES GENERALES

Atendiendo a los objetivos propuestos con esta investigación, se concluye que:

1- La revisión bibliográfica evidenció que la integración de datos es rica en cuanto a enfoques debido a la gran cantidad de información que es generada en la actualidad, lo cual permite realizar un sistema de integración de datos acorde a las necesidades específicas del problema.
2- El diseño e implementación de un método de integración de datos heterogéneos permite que sean analizados y procesados varios documentos a la vez e introducir todos sus datos en una ontología de dominio.
3- El análisis del experimento aplicado demostró que el método de integración de datos gubernamentales a una ontología de dominio reduce el tiempo de respuesta a la hora de integrar los datos de forma manual.

RECOMENDACIONES

La integración de datos va atada a las fuentes de datos en las que se trabajen. De manera que si se amplían las fuentes de entradas se podrá obtener un método más completo en cuanto a flexibilidad. Por tanto, se recomienda la implementación de un método que permita el trabajo con datos de diferentes fuentes a la vez o un esquema de alineación que transforme e integre los documentos en un mismo esquema general.

REFERENCIAS BIBLIOGRÁFICAS

ABELLO DIAZ, J.A., [sin fecha]. Obtener un método para la extracción de información a partir de documentos semiestructurados producidos al interior del Servicio Nacional de Aprendizaje SENA, permitiendo su publicación, reutilización e intercambio a través de la web semántica. S.l.: Universidad Nacional de Colombia-Sede Bogotá.

ALANI, H., KIM, S., MILLARD, D.E., WEAL, M.J., HALL, W., LEWIS, P.H. y SHADBOLT, N.R., 2003. Automatic ontology-based knowledge extraction from web documents. IEEE Intelligent Systems, vol. 18, no. 1, pp. 14-21.

ANGUITA SANCHEZ, A., 2012. Modelo de mediación semántica para la integración de fuentes de datos heterogéneas. S.l.: Informática.

BERNERS-LEE, T., HENDLER, J. y LASILA, O., 2001. La red semántica. Investigación y Ciencia__jul,

BERNERS-LEE, T., HENDLER, J. y LASSILA, O., 2001. The semantic web. Scientific american, vol. 284, no. 5, pp. 28-37.

BIZER, C., HEATH, T. y BERNERS-LEE, T., 2009. Linked data-the story so far. Semantic services, interoperability and web applications: emerging concepts, pp. 205-227.

BLANDÓN ANDRADE, J.C. y ZAPATA JARAMILLO, C.M., 2017. Una revisión de la literatura sobre población de ontologías. Revista Científica Ingeniería y Desarrollo, vol. 36, no. 1, pp. 259-284.

CRUZ, C. y NICOLLE, C., 2008. Ontology Enrichment and Automatic Population From XML Data. ODBIS, vol. 2008, pp. 17-20.

DANGER, R. y BERLANGA, R., 2009. Generating complex ontology instances from documents. Journal of Algorithms, vol. 64, no. 1, pp. 16-30.

DANIEL J. WEITZNER, HAROLD ABELSON, TIM BERNERS-LEE, JOAN FEIGENBAUM, JAMES HENDLER, AND GERALD JAY SUSSMAN, 2007. Information Accountability. ,

DATA, L.O., 2015. essentials. Center for American Progress,

DE KLERK, P.R., 2011. Linked open government data. ,

DOAN, A., HALEVY, A. y IVES, Z., 2012. Principles of data integration. S.l.: Elsevier. ISBN 0-12-391479-5.

EDEKI, C., 2013. Agile unified process. International Journal of Computer Science, vol. 1, no. 3.

GARANINA, N., SIDOROVA, E., KONONENKO, I. y GORLATCH, S., 2017. USING MULTIPLE SEMANTIC MEASURES FOR COREFERENCE RESOLUTION IN ONTOLOGY POPULATION. International Journal of Computing, vol. 16, no. 3, pp. 166-176.

GILLANI, S. y KO, A., 2015. Incremental ontology population and enrichment through semantic-based text mining: an application for it audit domain. International Journal on Semantic Web and Information Systems (IJSWIS), vol. 11, no. 3, pp. 44-66.

GRUBER, T.R., 1993. A translation approach to portable ontology specifications. Knowledge acquisition, vol. 5, no. 2, pp. 199-220.

GUERROUJ, L., 2013. Normalizing source code vocabulary to support program comprehension and software quality. Proceedings of the 2013 International Conference on Software Engineering. S.l.: IEEE Press, pp. 1385-1388. ISBN 1-4673-3076-0.

HEATH, T. y BIZER, C., 2011. Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology, vol. 1, no. 1, pp. 1-136.

HEREDIA, J., ÁLVAREZZA, L. y LINARES, N., 2011. Comparación y tendencias entre metodologías ágiles y formales. Metodología utilizada en el Centro de Informatización para la Gestión de Entidades. Revista Serie Científica, vol. 4, no. 10.

HUIJBOOM, N. y VAN DEN BROEK, T., 2011. Open data: an international comparison of strategies. European journal of ePractice, vol. 12, no. 1, pp. 4-16.

JANSSEN, M., CHARALABIDIS, Y. y ZUIDERWIJK, A., 2012. Benefits, adoption barriers and myths of open data and open government. Information systems management, vol. 29, no. 4, pp. 258-268.

KAIYA, H. y SAEKI, M., 2006. Using domain ontology as domain knowledge for requirements elicitation. Requirements Engineering, 14th IEEE International Conference. S.l.: IEEE, pp. 189-198. ISBN 0-7695-2555-5.

KNIBERG, H., 2015. Scrum and XP from the Trenches. S.l.: Lulu. com. ISBN 1-329-22427-2.

LENZERINI, M., 2002. Data integration: A theoretical perspective. Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. S.l.: ACM, pp. 233–246.

MAHMOOD, J. y REDDY, Y.R., 2014. Automated refactorings in Java using IntelliJ IDEA to extract and propogate constants. Advance Computing Conference (IACC), 2014 IEEE International. S.l.: IEEE, pp. 1406-1414. ISBN 1-4799-2572-1.

MCGUINNESS, D.L., FIKES, R., RICE, J. y WILDER, S., 2000. The chimaera ontology environment. AAAI/IAAI, vol. 2000, pp. 1123-1124.

MORENO, J.C. y MARCISZACK, M.M., 2013. La Usabilidad Desde La Perspectiva De La Validación de Requerimientos No Funcionales Para Aplicaciones Web. Córdoba: Universidad Tecnológica Nacional,

PASCA, M., 2004. Acquisition of categorized named entities for web search. Proceedings of the thirteenth ACM international conference on Information and knowledge management. S.l.: ACM, pp. 137-145. ISBN 1-58113-874-1.

PINKEL, C., BINNIG, C., JIMÉNEZ-RUIZ, E., MAY, W., RITZE, D., SKJÆVELAND, M.G., SOLIMANDO, A. y KHARLAMOV, E., 2015. RODI: A benchmark for automatic mapping generation in relational-to-ontology data integration. European Semantic Web Conference. S.l.: Springer, pp. 21-37.

PRESSMAN, R.S., 2005. Software engineering: a practitioner's approach. S.l.: Palgrave Macmillan. ISBN 0-07-301933-X.

RUIZ-MARTÍNEZ, J.M., MIÑARRO-GIMÉNEZ, J.A., GUILLÉN-CÁRCELES, L., CASTELLANOS-NIEVES, D., VALENCIA-GARCÍA, R., GARCÍA-SÁNCHEZ, F., FERNÁNDEZ-BREIS, J.T. y MARTÍNEZ-BÉJAR, R., 2008. Populating ontologies in the

eTourism domain. Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on. S.l.: IEEE, pp. 316-319. ISBN 0-7695-3496-1.

SÁNCHEZ, T.R., 2014. Metodología de desarrollo para la Actividad productiva de la UCI. Habana: sn,

SAORÍN, T., 2012. Cómo linked open data impactará en las bibliotecas a través de la innovación abierta. Anuario ThinkEPI, vol. 6, pp. 288-292.

SUTHERLAND, J., 2015. Scrum: El nuevo y revolucionario modelo organizativo que cambiara tu vida. S.l.: Planeta. ISBN 84-08-13532-5.

UBALDI, B., 2013. Open government data: Towards empirical analysis of open government data initiatives. OECD Working Papers on Public Governance, no. 22, pp. 0_1.

VILLAR, L.D., CASTRO, K.C. y BERMÚDEZ, G.M.T., 2017. Datos abiertos y su beneficio en la contratación pública. Redes de Ingeniería, pp. 80-90.

YOON, H.-G., HAN, Y.J., PARK, S.-B. y PARK, S.-Y., 2007. Ontology population from unstructured and semi-structured texts. Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on. S.l.: IEEE, pp. 135-139. ISBN 0-7695-2930-5.

BERNERS-LEE, T. (2006). Linked Data - Design Issues. Retrieved July 23, http://www.w3.org/DesignIssues/LinkedData.htm

ISO, I., 2011. IEEE. 29148: 2011-Systems and software engineering-Requirements engineering. S.l.: Technical report.