

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ**

**IMPLEMENTACIÓN DE UN SOFTWARE PARA BÚSQUEDA DE
PUBLICACIONES CIENTÍFICAS EN BASES DE DATOS
ESTRUCTURADAS MEDIANTE DATOS ENLAZADOS**

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

Iván Renato Uribe Canchanya

ASESOR: Dr. Héctor Andrés Melgar Sasieta

Lima, febrero del 2018

Dedicatoria

A Dios por estar siempre presente en este camino de la vida.

A mis padres porque son la piedra angular de cada uno de mis logros.

A mis hermanos porque son un impulso extra en cada uno de mis pasos.



Agradecimientos

A Dios, mis padres y hermanos por el apoyo incondicional y por estar siempre presentes.

A mis profesores que a lo largo de la carrera ayudaron a mi crecimiento.

A todo aquél que contribuyó beneficiosamente en esta etapa de mi vida: amigos, mentores...



RESUMEN

En la actualidad, la información es uno de los activos más importantes tanto en niveles personales, educativos y organizacionales. La información permite el desarrollo y el avance de los estados del conocimiento. Desde la aparición del Internet y su exponencial evolución, el acceso a la información se ha vuelto universal y su cantidad disponible sigue aumentando considerablemente.

Para las organizaciones es muy valioso el resguardo y uso de la información ya que, de esta administración depende su capacidad para seguir creciendo y obteniendo valor dentro de sus respectivos campos de acción. Del mismo modo, para una persona el obtener información importante es adquirir conocimientos que serán relevantes para la consecución de sus objetivos planteados. En cualquiera de los ámbitos el uso de la información permite la formulación y la ejecución de los planes estratégicos. Sin embargo, el encontrar aquella información que realmente se necesita se ha vuelto una tarea cada vez más complicada.

En el campo de la investigación científica la recopilación de fuentes de información representa el punto de partida. Trasladando esta necesidad al contexto peruano, se observa que la producción científica está en aumento y en un ritmo acorde al crecimiento de otras naciones. Sin embargo, actualmente resulta complicado para los investigadores el obtener la información relevante para iniciar sus investigaciones con la certeza de que no existen investigaciones previas o que abarquen el mismo campo de estudio. Para lograr una buena recopilación se debe navegar entre los distintos repositorios digitales existentes que son de libre acceso o incluso pagados.

En tal sentido, existe una solución conocida como Datos Enlazados, un enfoque que no se contrapone a la web actual que permite el vínculo entre páginas web (documentos HTML), pero sí lo complementa ya que permite la vinculación de datos entre distintos contextos y fuentes de información. El presente trabajo de fin de carrera propone una alternativa de solución para la

búsqueda integral y automatizada en distintos repositorios digitales que son de libre acceso y cuyas bases de datos están estructuradas mediante Datos Enlazados.

Se implementó un método de búsqueda adaptativo en base a las ontologías que manejan los distintos repositorios digitales (datasets). De tal manera, se obtiene una ontología de dominio adaptable que permite la extracción de datos relevantes de cada repositorio, para su posterior reestructuración y su visualización. Para lograr la ontología dinámica se implementó un algoritmo adaptador que analiza el vocabulario ontológico del dataset e identifica las entidades relevantes para el dominio de investigaciones y publicaciones. Luego, se implementaron métodos de extracción con funciones en SPARQL que dependen de la ontología y finalmente, los datos relevantes son guardados en grafos RDF para luego ser serializados en documentos RDF/XML y Turtle.

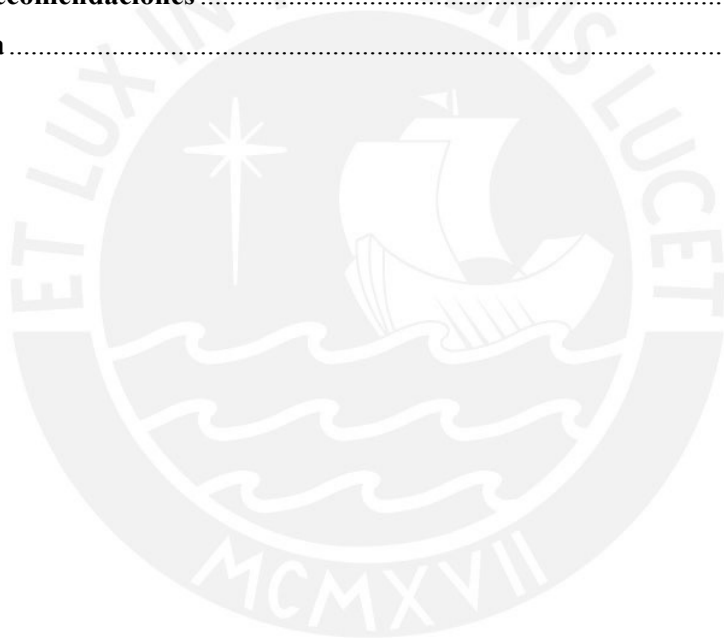
Se concluye que el proyecto ha sido exitoso en cuanto que el software permite realizar la búsqueda de publicaciones de distintos autores peruanos que tienen indexados sus documentos en repositorios digitales como DBLP o IEEE Library Project, contribuyendo de esta manera a la búsqueda integral de información.

Contenido

CAPÍTULO 1: Generalidades y definición del Proyecto	1
1.1 Problemática	1
1.2 Estado del Arte	9
1.2.1 Objetivos del Estado del Arte	9
1.2.2 Método y planteamiento de la revisión	9
a) Etapa 1: Planeamiento de la revisión	9
b) Etapa 2: Ejecución de la revisión	10
1.2.3 Estudios Seleccionados	12
a) E1: Open CSDB (Open Chinese Science Data Base)	12
b) E2: Europea	15
c) E3: Open Archives Initiative Protocol for Metadata Harvesting (OIA-PMH) 16	
d) E4: Open Science	17
1.2.4 Discusión	19
1.2.5 Conclusiones del Estado del Arte	21
1.3 Marco Conceptual	21
1.3.1 Ontología	21
1.3.2 <i>Linked Data</i>	21
a) Uso de URI	24
b) Uso de URI HTTP	25
c) Resource Description Framework (RDF)	26
d) Incluir enlaces a otras URI	27
1.3.3 La Web clásica y la Web de los datos	28
1.3.4 Metadatos	29
1.3.5 Datasets	30
CAPÍTULO 2: Objetivos y Resultados Esperados	33
2.1 Objetivo General	33
2.2 Objetivos Específicos	33
2.3 Resultados Esperados	33
2.4 Herramientas, Métodos y Metodologías	34
2.4.1 Herramientas	35
2.4.2 Metodologías	37
2.5 Alcance y Limitaciones	38
2.5.1 Riesgos	38

2.6	Justificación de la solución	39
2.7	Análisis de Viabilidad	40
2.7.1	Viabilidad técnica.....	40
2.7.2	Viabilidad temporal.....	40
2.7.3	Viabilidad económica.....	41
2.7.4	Análisis de necesidades	41
CAPÍTULO 3: Diseño de la ontología dinámica		42
3.1	Diseño metodológico de la ontología.....	42
3.1.1	Determinar el dominio y alcance de la ontología base.....	42
3.1.2	Considerar la reutilización de ontologías existentes	43
3.1.3	Enumerar términos importantes de la ontología	45
3.1.4	Definir las clases y jerarquías de clases.....	45
3.1.5	Definir las propiedades de las clases	45
3.1.6	Definir las restricciones de las propiedades.....	46
3.2	Resultado de la Metodología	47
3.3	Validación de la ontología	48
3.4	Algoritmo adaptador de ontología.....	48
3.4.1	Pseudocódigo principal.....	48
3.4.2	Leer Ontologías	49
3.4.3	Extraer Recursos.....	50
3.4.4	Generar Ontología	52
3.4.5	Serializar.....	52
3.4.6	Generación y mantenimiento del corpus de palabras claves.....	53
CAPÍTULO 4: Diseño de métodos de extracción		59
4.1	Pseudocódigo Principal.....	59
4.2	Pseudocódigo Obtener URI Autor.....	60
4.3	Pseudocódigo Obtener Tuplas	63
4.4	Pseudocódigo Obtener URI Publicaciones	64
CAPÍTULO 5: Diseño de métodos de transformación		67
5.1	Pseudocódigo Principal.....	67
CAPÍTULO 6: Diseño de la interfaz web		69
6.1	Metodología para la gestión y desarrollo del proyecto.....	69
6.2	Requisitos del Software	69
6.3	Análisis.....	70
6.4	Arquitectura de Software.....	70

6.5	Prototipos	71
6.5.1	Mantener Usuarios.....	72
6.5.2	Mantener Autores.....	74
6.5.3	Consultar Publicaciones.....	76
6.5.4	Generar Estadísticas.....	79
CAPÍTULO 7: Conclusiones		80
7.1	Escenario de pruebas y resultados.....	80
7.2	Observaciones del proyecto.....	84
7.3	Conclusiones.....	85
7.3.1	Conclusiones sobre los objetivos y resultados esperados.....	85
7.3.2	Conclusiones Generales.....	86
7.4	Recomendaciones.....	87
Bibliografía		88



CAPÍTULO 1: Generalidades y definición del Proyecto

En el presente capítulo se aborda la problemática de la producción científica en el Perú. Asimismo, se evalúan soluciones implementadas en distintos lugares para ayudar a la gestión de las publicaciones científicas y académicas. Por último, se mencionan los conceptos principales que permiten el entendimiento del presente proyecto.

1.1 Problemática

La producción científica peruana no está al nivel de los principales países de la región y aún se encuentra por debajo del promedio de Latinoamérica; sin embargo, existen indicadores importantes que evidencian el crecimiento de las publicaciones de investigación científica y académica en el Perú (CONCYTEC, 2014). En tal sentido, en el período de 1996 al 2011 la cantidad de artículos se quintuplicó de 164 publicaciones a un total de 1,116 publicaciones (CONCYTEC, 2014). Esa tendencia incluso ha sido mayor en un segundo período del 2011 al 2015, en el cual la producción científica pasó a tener 14,434 publicaciones de acuerdo con la *Tabla 1.1*, lo cual deviene en un aumento en cantidad de doce veces en comparación al 2011.

	País	Documentos	Documentos citables	Citas	Autocitas	Citas por Documento	Índice H
1	United States	9360233	8456050	202750565	94596521	21.66	1783
2	China	4076414	4017123	24175067	13297607	5.93	563
3	United Kingdom	2624530	2272675	50790508	11763338	19.35	1099
4	Germany	2365108	2207765	40951616	10294248	17.31	961
5	Japan	2212636	2133926	30436114	8352578	13.76	797
			●				
			●				
			●				
71	Cyprus	17072	15592	172117	20409	10.08	127
72	Latvia	16350	15851	119627	17472	7.32	112
73	Iceland	15625	14353	357678	32540	22.89	218
74	Sri Lanka	14434	13201	192443	20509	13.33	154
75	Puerto Rico	13641	13293	248888	15917	17.98	166

Tabla 1.1 Estadísticas de publicaciones de SCImago Journal and Country Rank (SCImago Journal, 2016)

Otro dato importante es la cantidad de publicaciones citables que abarca un 92% del total -en el caso de USA es del 91%-. Una publicación citable es aquella que ha pasado por un proceso de revisión que acredita su contribución a la literatura científica y, por lo tanto, tiene las siguientes características: está presente en formato digital (alojada en algún repositorio o revista científica), puede ser accedida (referenciada) mediante un único Identificador de Recursos Uniforme (URI, por sus siglas en inglés) evitando cualquier tipo de ambigüedad y estará disponible a lo largo del tiempo (Blakley, 2010). Esta medición indica la calidad de las publicaciones peruanas que cumplen con los estándares de calidad internacionales para que puedan ser indizadas en distintas bibliotecas digitales.

Sin embargo, existen pocos esfuerzos para que dichas publicaciones estén registradas en un repositorio académico propio. Uno de esos esfuerzos es SCielo Perú¹, un proyecto que brinda una biblioteca virtual de revistas científicas peruanas y que es liderado por el Consejo Nacional de Ciencia, Tecnología e Innovación (CONCYTEC). SCielo Perú tiene que ser actualizado en su contenido ya que no posee vínculos directos o no indizan los artículos y publicaciones de las revistas digitales especializadas (SCielo Perú, 2016) -páginas web científicas- como sí lo hacen muchas otras bases de datos digitales como Scopus, ProQuest, IEEE/IET Electronic Library o Digital Bibliography and Library Project (DBLP).

Asimismo, otro proyecto que también es liderado por CONCYTEC es ALICIA² (Acceso Libre a la Información Científica), el cual es el repositorio nacional digital implementado en Diciembre 2013 (Melgar Sasieta, 2017). ALICIA brinda acceso libre a información intelectual producida por entidades del sector público o financiadas por el Estado, así como del sector privado o personas naturales que deseen compartir su información ("Manual de uso de ALICIA", s. f.). Nuevamente el problema radica en que solo muestra contenido de su propio repositorio -el cual debe ser gestionado y actualizado-, aunque sí permite la adhesión de repositorios de instituciones que sigan el procedimiento y las directivas del Reglamento de la Ley 30035 según se muestra en la *Figura 1.1*.

Por otro lado, la gran mayoría de publicaciones científicas peruanas forma parte de distintas bases de datos digitales, en las cuales las publicaciones son indizadas en caso cumplan con los criterios de calidad exigidos internacionalmente. La complejidad radica en que cada una de estas bibliotecas digitales posee sus propias estructuras de datos que dificultan una búsqueda estandarizada y muchas veces el acceso no es total debido a los derechos de autor o a que son bibliotecas digitales privadas que poseen un costo elevado.

¹ <http://www.scielo.org.pe/>

² <https://alicia.concytec.gob.pe/vufind/>

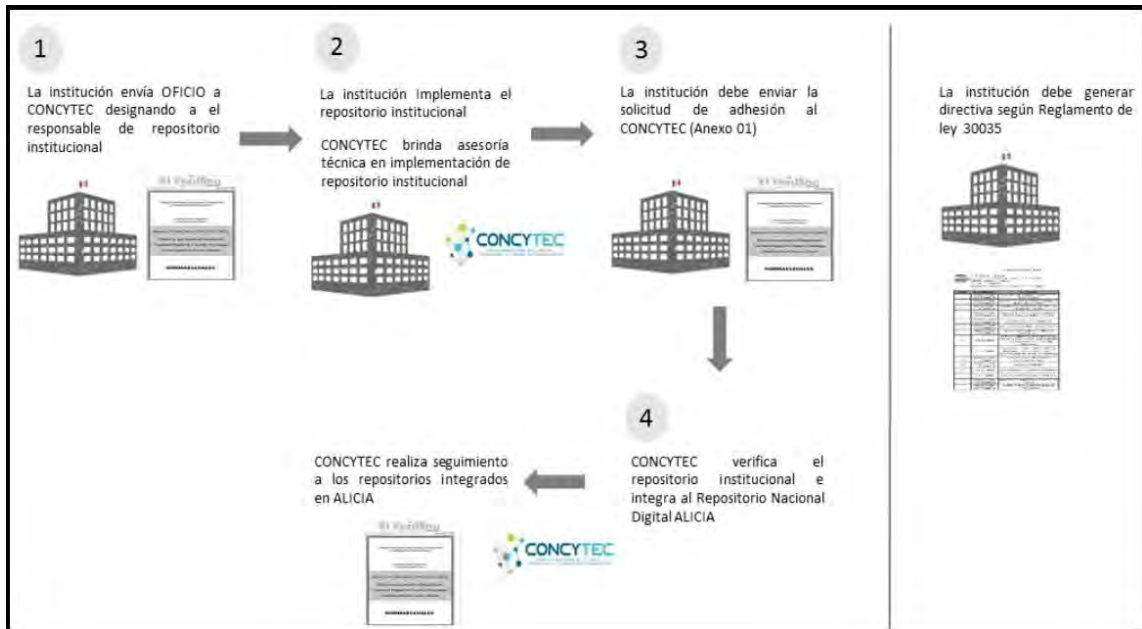


Figura 1.1 Procedimiento de Adhesión a ALICIA ("Requisitos para adherirse al repositorio Nacional - Manual de uso de ALICIA", s. f.)

De las bases de datos mencionadas anteriormente, la DBLP publica sus estructuras de datos mediante el método de los Datos Enlazados o *Linked Data* (LD) (Albert-Ludwigs-Universität Freiburg, 2009). De tal manera, provee servicios web de datos abiertos o de libre acceso (DBLP, 2016). Un *dataset* es una base de datos estructurada bajo el método de los Datos Enlazados; este método permite vincular los datos de distintas fuentes de información y, en este caso, de distintas áreas y temas de investigación científica y académica (Miao, Meng, Fang, Nishino, & Igata, 2015). Asimismo, los datos también pueden vincularse entre distintos *datasets*: formando así un espacio global de datos que se vinculan entre sí. De acuerdo a la *Figura 1.2* se observa un esquema de grafos que representa la vinculación de los datos en la DBLP. En tal sentido, la base de datos de la DBLP también puede ser catalogada como un *dataset*.

La red de LD está en constante crecimiento y hoy representa la forma en que el contenido más granular de la información -los datos- puede ser tratado de manera estructurada y estandarizada para que sea accedida mediante un mecanismo uniforme (W3C, s. f.). Esta tecnología amplía las capacidades de la Web actual -que sólo permite la vinculación entre documentos o páginas web- y en la actualidad es ampliamente usada por bibliotecas digitales que han adaptado sus bases de datos debido a las potencialidades que brinda para el manejo de los datos (Peset, Ferrer, & Subirats, 2011).

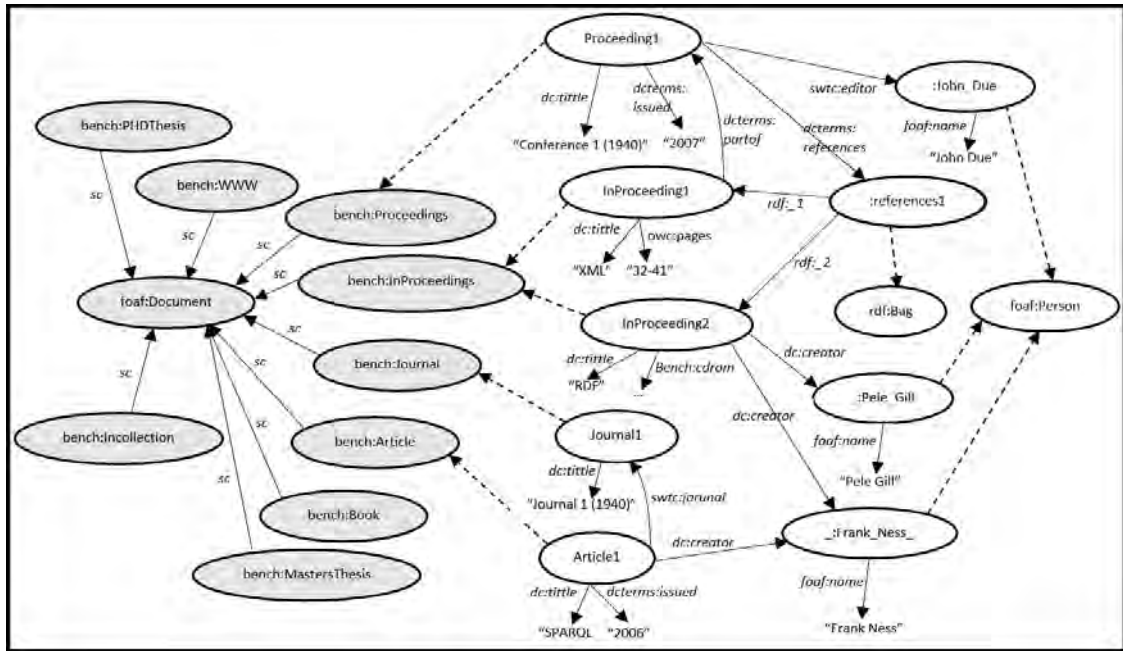


Figura 1.2 Estructura de datos de DBLP (Albert-Ludwigs-Universität Freiburg, 2009)

A pesar de las ventajas mencionadas, los *datasets* no son accesibles en su totalidad ya que, de igual manera a lo mencionado anteriormente, algunos pueden ser privados o existen restricciones por derechos de autor, por lo que son limitados en cuanto a la información a la que se puede acceder libremente. Sin embargo, existen proyectos para que la red de LD sea abierta en su totalidad. El conjunto de estos proyectos conforma la red de los Datos Abiertos Enlazados o Linked Open Data (LOD) que sí pone a disponibilidad la totalidad de la información para el acceso público (Peset et al., 2011). En tal sentido, existe un conjunto de herramientas y definiciones que permite el manejo de LOD.

Por un lado, existen definiciones de tipos y estructuras en base a reglas formales para la descripción de los datos en cuanto a sus propiedades y relaciones o vínculos con otros; estas definiciones también son llamadas Ontologías y es en base a esta que la información es organizada. El formato o estructura común a todas las ontologías es el Resource Description Framework (RDF³) (Guha, Brickley, & Macbeth, 2016), el cual es el estándar de utilización de grafos para la descripción de los recursos (datos) mediante la relación sujeto-predicado-objeto (E. Méndez Rodríguez, 1999). El sujeto define al recurso, el predicado define al atributo o propiedad y el objeto corresponde al valor de dicho atributo. Este comportamiento es similar al de las bases de datos relaciones (BDR) y al de la programación orientada a objetos (POO).

³ <https://www.w3.org/RDF/>

Asimismo, el predicado y el objeto también pueden tratarse de otros recursos; por lo tanto, lo que se produce es una relación entre recursos o datos. Y, al igual que en las BDR, existe un método de consultas para el acceso y el manejo de estas estructuras de datos llamado SPARQL, el cual es el lenguaje de consultas de grafos RDF. La interfaz para la ejecución de dichas consultas es conocida como *SPARQL Endpoint* y, en este sentido, cada *dataset* puede poseer su propia interfaz para el manejo de sus propias ontologías estructuradas bajo grafos RDF (Mateus, Ruiz, & Plaza, 2015). El inconveniente es que estas interfaces solo pueden ser usadas por usuarios que poseen los conocimientos técnicos de SPARQL y que saben cómo se han definido las ontologías.

Si bien el uso de LD permite un acceso a los datos más preciso y métodos de búsqueda más óptimos (Peis et al., 2003), un nuevo inconveniente es la gran cantidad de *datasets* que manejan sus propias ontologías de acuerdo a los campos de investigación que abarcan (Ríos, Martín, & Ferreras, 2012). Una solución a nivel de usuario es el uso de los buscadores que dichos *datasets* también brindan y que permiten búsquedas especializadas mezclando distintos campos como título, autor, palabras claves en el contenido, entre otros. De esta manera, la consulta resulta sencilla para un usuario común sin los conocimientos de SPARQL y ontologías.

En base a lo presentado hasta el momento, se han encontrado aspectos positivos en el uso de LD en comparación de lo que ofrece la web actual, pero también existen inconvenientes que no se han podido minimizar como es el caso de la proliferación de *datasets* con sus propias ontologías y que para acceder a sus fuentes de información se tiene que acceder a cada uno de sus buscadores o interfaces de manera individual para hacer uso de los recursos. Dichas ventajas y desventajas se presentan en la *Tabla 1.2*.

Siendo la bibliometría el análisis estadístico del tamaño, crecimiento y distribución de la bibliografía (Piñero López, 1972), para poder hacer una radiografía bibliométrica de las publicaciones peruanas se tendría que acceder a cada uno de los *datasets* especializados en publicaciones científicas y académicas para realizar las búsquedas de los artículos cuyos autores sean de origen peruano. Por lo tanto, en la actualidad no existe un mecanismo que permita tener una base de datos unificada a excepción del proyecto de Scielo o ALICIA -mencionados previamente- cuyo limitante radica en que está formado por su propia base de información y que no se encuentra enlazada con otros *datasets* de páginas web científicas o revistas digitales (Scielo Perú, 2016).

Ventajas	Desventajas
Se encuentra información relevante de manera más eficiente	La gran cantidad de documentación que ya está publicada en la web tiene que ser reestructurada al formato propuesto por <i>Linked Data</i>
Permite compartir la información de forma más sencilla	Gran cantidad de <i>datasets</i> con sus propias ontologías definidas que pueden generar ambigüedad. Por ejemplo: un mismo científico puede ser llamado de distintas formas en dos <i>datasets</i> o incluso dentro del mismo. En uno puede ser nombrado como “Pedro Paulet” y en otro como “Paulet Mostajo, Pedro”(DBPEDIA, s. f.)
Mayor facilidad para realizar modificaciones en el diseño de los datos dado que las estructuras siguen un estándar	Tecnología más compleja para el usuario común
Mejor uso de los recursos presentes en la web tanto para las personas como para las máquinas	

Tabla 1.2 Ventajas y desventajas de *Linked Data*

En tal sentido, con el aumento de las publicaciones académicas y científicas peruanas, la búsqueda de literatura científica peruana requiere de un mayor esfuerzo dado que finalmente terminan en distintos repositorios que indizan las publicaciones y que pueden ser de acceso libre o no. Si bien existen los repositorios Scielo Perú y ALICIA -siendo ALICIA el principal esfuerzo del Estado- que han permitido centralizar información intelectual, aún existe gran cantidad de producción científica peruana que termina alojada en bases de datos indizadas internacionalmente (Melgar Sasieta, 2017). Estas publicaciones de autores peruanos pasan por los procesos de revisión de los repositorios digitales internacionales que garantizan la calidad científica de la publicación. Entonces, se tiene producción científica de calidad internacional a la que no se puede acceder de manera rápida y eficaz desde un buscador integral.

En concordancia a lo mencionado y según la *Figura 1.3*, el problema que se presenta es el déficit para poder centralizar en un solo repositorio las publicaciones peruanas de investigación científica y académica. Tal como se ha mencionado, el mayor esfuerzo del Estado se traduce en ALICIA. Sin embargo, hay una alta producción científica peruana que es indizada en repositorios internacionales. Una de las causas de dicho problema es que hay una mayor dificultad para acceder a la información porque se necesita usar los buscadores especializados para cada uno de los repositorios o porque su uso está limitado a un pago -sin la posibilidad de

obtener los metadatos y mucho menos la publicación-. Entonces, a lo que se enfrenta un investigador es a realizar extensivas búsquedas en cada una de las plataformas o repositorios para obtener información sobre las publicaciones -a veces obteniendo los mismos resultados a un costo mayor de tiempo- en un campo o área específico.

Asimismo, al analizar solamente aquellos repositorios que ya cuentan con metadatos estructurados mediante LD, se observa que cada uno usa su propia estructura u ontología. Lo que se origina es una ambigüedad de información, ya que distintos repositorios pueden contener la misma información de publicaciones que están relacionadas a alias de autores que difieren textualmente pero que se refieren al mismo conceptualmente. Peor aún, cada uno de los repositorios también cuenta con sus propios buscadores especializados para su dataset, ya sea para usuarios expertos -SPARQL EndPoint- o inexpertos -interfaz web de búsqueda-. La ventaja es que, a pesar de que sólo permiten la búsqueda de su propio dataset -tal como se realiza en un buscador de un repositorio no estructurado mediante LD-, los datos que se puedan encontrar podrían tener relación con datos cuyo origen está en un dataset distinto. Esta característica brinda una ventaja a los metadatos estructurados mediante LD en comparación de un documento común de la web (documento html) que presenta los metadatos como parte de su información.

Debido a esa ventaja, mediante el propio uso de LD se puede abordar el problema en tres etapas por medio del uso de las tecnologías y herramientas previamente mencionadas y que forman parte de los mecanismos que brinda LD. Estas etapas también son conocidas como procesos ETL (extracción, transformación y carga). Es decir, lo que se busca es la extracción de los datos de los distintos *datasets* de publicaciones académicas y científicas. Luego, en base al entendimiento de sus ontologías, los datos son estructurados (transformados) de manera estandarizada para permitir su integración. Por último, esta nueva estructura es presentada en formatos útiles para todo tipo de usuario (Takahiro Komamizu, Toshiyuki Amagasa, & Hiroyuki Kitagawa, 2016).

El presente proyecto de tesis busca hacer frente al problema o déficit existente para centralizar la información de las publicaciones académicas y científicas peruanas en un solo repositorio digital que permita tener registro de la actividad de los investigadores y que sirvan como base para las investigaciones futuras. También se pone en manifiesto la complejidad de la extracción de información en el contexto de los Datos Enlazados. En tal sentido, se presenta la viabilidad de una solución alternativa, la cual consiste en la búsqueda de información en distintos conjuntos de datos que actualmente existen (*datasets*), que son de libre acceso y que forman parte del proyecto de *Linking Open Data* (LOD).

En la *Figura 1.3* se presenta el árbol de problemas que resume y engloba cada una de las causas y efectos del problema a abordar en el presente trabajo de fin de carrera.

ARBOL DE PROBLEMAS	EFFECTOS	Desconocimiento de las investigaciones realizadas por autores peruanos	Mayor complejidad en la búsqueda de información y bajo nivel de competitividad internacional	Dificultad de realizar nuevas investigaciones en base a otras ya realizadas	Desmotivación de los investigadores y redundancia en las investigaciones
	PROBLEMA	Existe un déficit para registrar o buscar de forma automática, integral y centralizada las publicaciones peruanas de investigación científica y académica			
	CAUSAS	Para acceder a los metadatos de las publicaciones los usuarios deben consultar en distintos repositorios digitales donde se encuentran indizados dichos documentos científicos	Los distintos <i>datasets</i> (bases de datos) de los repositorios forman parte de la nube de Linked Data y no todos son de libre acceso, teniendo cada uno sus propios mecanismos de búsqueda de información de las publicaciones	Los repositorios usan sus propios estándares (estructuras y ontologías) para la representación de la información tanto para los metadatos como para los datos de las publicaciones, provocando ambigüedad al definir los mismos conceptos (recursos) de distintas formas	Cada repositorio digital cuenta, de manera individual, con interfaces que brindan servicios a usuarios finales expertos o inexpertos: SPARQL EndPoint e interfaces web de buscadores respectivamente; los cuales están limitados sólo a la búsqueda de su propio <i>dataset</i>

Figura 1.3 Árbol de Problema

1.2 Estado del Arte

Mediante el análisis del Estado del arte se podrán reconocer proyectos relevantes en el ámbito de la producción científica que han brindado soluciones a escala o a medida según los problemas encontrados en su propio contexto.

1.2.1 Objetivos del Estado del Arte

El objetivo de la revisión del estado del arte es reconocer lo que ya se ha realizado sobre el área en cuestión y si existen soluciones a la problemática de las publicaciones académicas y científicas peruanas. De tal manera, se puede determinar el aporte del presente proyecto brindando una alternativa para la resolución del problema.

1.2.2 Método y planteamiento de la revisión

El presente proyecto utilizará el método de revisión sistemática, el cual permitirá la identificación, recolección, validación e interpretación de las investigaciones relevantes para los propósitos del problema a abordar (Kitchenham, 2004) (Kitchenham, 2007). Como parte de la revisión se tomaron las siguientes consideraciones:

a) *Etapa 1: Planeamiento de la revisión*

- **Enfoque:** identificar proyectos que hayan abordado o resuelto el problema de las publicaciones dispersas en distintos formatos mediante el uso de Linked Data. Se procurará verificar distintas soluciones de los distintos continentes del mundo.

- **Preguntas:**
 - ¿Cuáles son los problemas o limitaciones del método clásico para estructurar la información en las bibliotecas y repositorios digitales?
 - ¿Cuáles fueron las necesidades y objetivos para la búsqueda de una solución al problema?
 - ¿Cuáles fueron las acciones e iniciativas para la resolución del problema?
 - ¿Qué tecnologías se usaron para el planteamiento e implementación de la solución?
 - ¿Cuáles fueron los inconvenientes y desafíos para ejecutar la solución planteada?

- Las formulaciones de las preguntas están basadas en el método PICO (Mark & Helen, 2006):

- **Población:** Soluciones al problema de integración de las publicaciones científicas de distintos repositorios digitales
 - **Intervención:** Soluciones basadas en la aplicación de *Linked Data*
 - **Contexto:** Este estudio abarcará a instituciones educativas, bibliotecas y repositorios digitales, asociaciones y redes de investigación científica
- **Criterios de inclusión o exclusión:** El resultado de la búsqueda debe ser un compendio de artículos y documentos que no sobrepasen los 10 años de antigüedad y optando preferentemente por aquellas fuentes publicadas desde el 2013 en adelante.

b) Etapa 2: Ejecución de la revisión

- **Palabras claves y términos de búsqueda:** En base al método de búsqueda se seleccionaron palabras claves relacionadas a la temática y contexto del problema:
- *Linked Data*
 - *Publishing*
 - *ETL*
 - *SPARQL*
 - *Academic*
 - *Ontologies*

Estas palabras conformarán las cadenas de búsqueda.

- **Estrategia de búsqueda:** La búsqueda de fuentes será realizada en las siguientes bases de datos digitales: *Scopus*, *IEEE Xplore Digital Library*, *ProQuest*, *ACM Digital Library* y *Data Science Journal*. Asimismo, se optarán con otras alternativas como *Google Scholar* y la búsqueda abierta en Internet de soluciones de instituciones públicas como CONCYTEC.
- **Ejecución de la búsqueda:**
- **Scopus:** “*Linked Data*” AND “*Publishing*”, “*Linked Data*” AND “*Academic*” AND “*Ontologies*”, “*Linked Data*” AND “*ETL*”, “*ETL*” AND “*SPARQL*”
 - **ProQuest:** “*Linked Data*” AND “*Publishing*” como título del documento
 - **IEEE Xplore Digital Library:** “*Linked Data*” AND “*Publishing*” AND “*Academic*”
 - **Data Science Journal:** “*Linked Data*”, “*Linked Data*” AND “*Academic*”

- **Otros:** mediante el Google Scholar se realizarán búsquedas con las mismas cadenas descritas, así como la búsqueda en las publicaciones del CONCYTEC.

➤ **Resultados de Búsqueda:**

- **Scopus:**
 - *Towards linked research data: An institutional approach.* Esta solución fue escogida debido a su contexto institucional.
 - *Linked data platform D2R+.* Este artículo explica la instalación y despliegue del servidor D2R para implementar una solución basada en *Linked Data*. Se descartó su utilización ya que su enfoque estaba fuera del contexto.
- **ProQuest:**
 - *Publishing Chinese medicine knowledge as Linked Data on the Web.* Se presenta una solución basada en *Linked Data* para la publicación de datos sobre medicina China. No se tomará en cuanto dado que el dominio de la solución es distinto al de la problemática planteada. Sin embargo, presenta información relevante que puede ser usada en otros apartados.
 - *Metadata management, interoperability and Linked Data publishing support for Natural History Museums.* Se presenta la metodología para que un conjunto de repositorios digitales del Museo de Historia Natural pueda realizar la transición a Web Semántica y publicar los metadatos como *Linked Data*. Este caso consiste en asimilar dichos repositorios a Europeana, la biblioteca digital que abarca a todos los países de la Unión Europea. Se selecciona este artículo en conjunto con el caso Europeana (buscado en Internet).
- **IEEE Xplore Digital Library:**
 - *Integrating linked sensor data for on-line analytical processing on-the-fly.* Presenta una solución orientada a la publicación de información obtenida mediante sensores, cuyo dominio está fuera del objeto de la revisión; por tanto, se descarta.
- **Data Science Journal:**
 - *Opensdb: Research on The Application of Linked Data in Scientific Databases.* Fue escogido debido a que especifica claramente un método de

solución para la integración de bases de datos de repositorios digitales mediante el uso de *Linked Data*.

○ **Internet:**

- Aparte de realizar las búsquedas con los términos en inglés, se realizó el mismo proceso en castellano y como resultado se encontró una publicación en el contexto latinoamericano: “**Una aproximación basada en linked data para la integración de repositorios digitales abiertos latinoamericanos**”. Fue escogido debido a la aproximación del contexto para el caso peruano, además que proveía un método completo de solución de acuerdo a los objetivos de la revisión sistemática.
- Se encontró en el Repositorio Español de Ciencia y Tecnología (RECYT) la siguiente publicación: “**Linked Data y Linked Open Data: su implantación en una biblioteca digital. El caso Europeana**”. Se escoge esta publicación dado que describe a la red de bibliotecas digitales más grande de Europa y su impacto en la adaptación a *Linked Data*.

- **Criterios de inclusión o exclusión:** Fueron seleccionados solamente aquellos estudios cuyas soluciones estaban basadas en *Linked Data* con el objetivo de unificar repositorios digitales de publicaciones académicas y científicas. Después de la aplicación de los criterios de inclusión y exclusión se seleccionaron 4 estudios relevantes, cuyas soluciones fueron aplicadas en Latinoamérica, Europa y Asia.

1.2.3 Estudios Seleccionados

En esta Sección se presentan los estudios seleccionados que tuvieron un mayor impacto mediante las soluciones planteadas.

a) E1: Open CSDB (Open Chinese Science Data Base)

El proyecto Open CSDB introdujo una aplicación cuya estructura de datos fue implementada bajo el método de Datos Enlazados. De esta manera se abordó el problema de las bases de datos científicas que habían acumulado más de 200TB de información científica. Dichas bases de datos relacionales consistían en 2 bases de datos de referencia, 8 bases de datos temáticas, 4 bases de datos de temas especiales y 37 bases de datos especializadas. Cada base de datos poseía su propia estructura para el almacenamiento y la organización de la información, así como sus

propios servicios web que proveían los accesos necesarios para el uso de la información por parte de los usuarios finales.

Del mismo modo, las soluciones implementadas eran heterogéneas con distintos ambientes de servicio y niveles, un tratamiento distinto en la semántica y relevancia de los datos, así como sus propios mecanismos de seguridad. Debido a estos inconvenientes se requería un mecanismo de acceso a los datos que fuera inclusivo, universal, asociado a un soporte semántico, descentralizado y de bajo costo (Shen, Li, & Han, 2015). La solución aplicó los principios de Linked Data para establecer un estándar en el cual se pudieran adaptar todas las bases de datos, este estándar consiste en lo siguiente:

- Cada base de datos tiene una única URI.
- Cada registro de la base de datos posee una única URI.
- Cada petición de las URI HTTP devuelve ya sea páginas web o datos estructurados mediante RDF.
- Cada base de datos tiene publicada sus vocabularios RDF para acceder a sus datos.
- Además de las propiedades de los datos, se conocerán las diferencias semánticas entre los datos.
- Cada base de datos posee una interfaz SPARQL Endpoint para las consultas de los grafos RDF.

La *Figura 1.4* muestra el resultado de la conversión de un registro a un grafo RDF basado en la terna de sujeto-predicado-objeto para describir cada campo del registro.

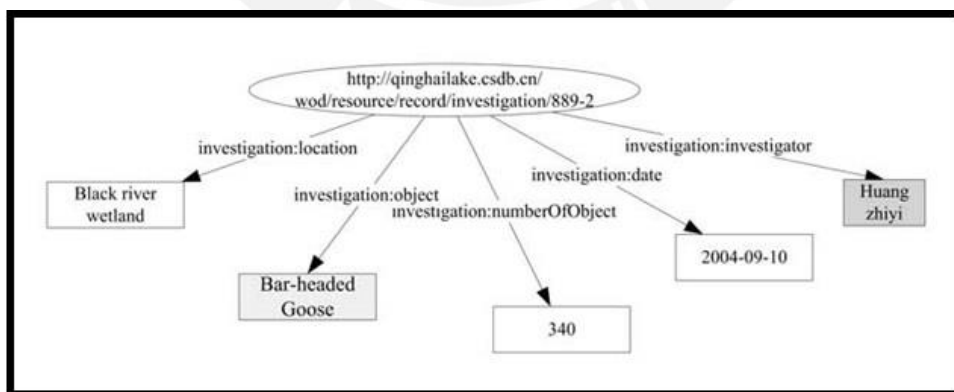


Figura 1.4 Conversión de un registro a grafo RDF (Shen et al., 2015)

Asimismo, la *Figura 1.5* muestra la arquitectura de la solución basada en tres capas: capa de aplicación, capa de administración y capa de construcción.

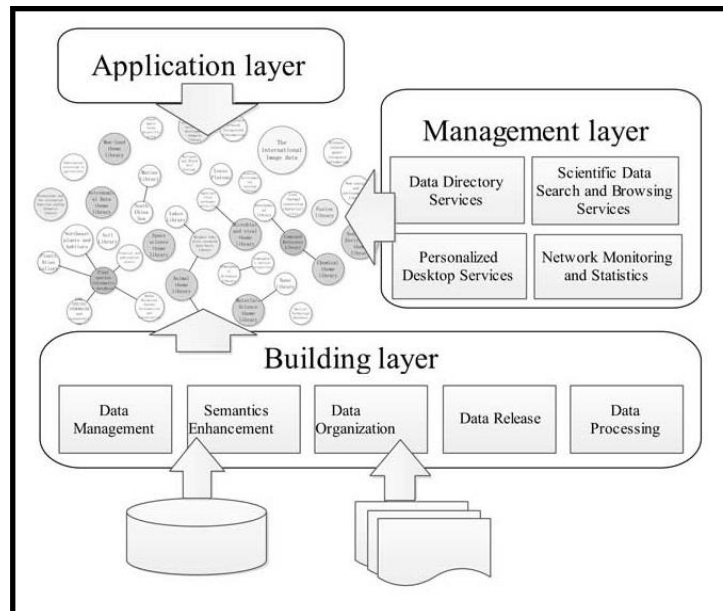


Figura 1.5 Arquitectura del Sistema (Shen et al., 2015)

La capa de construcción se encarga del enlazado de las distintas bases de datos mencionadas. Posee 5 características relevantes:

1. Administración de los datos: se encarga de estructurar los datos que no tenían un formato, así como de gestionar su ubicación.
2. Mejora Semántica: Incluye la extracción de los metadatos de los archivos.
3. Organización de los datos: es responsable de la clasificación y organización, así como del diseño de las URI y del vocabulario.
4. Liberación de datos: realiza el mapeo de las bases de datos relacionales a un modelo de datos RDF.
5. Procesamiento de datos: construye una base de datos temática a partir de las bases de datos que conforman la red.

La capa de gestión contiene los servicios y la administración de la red de datos. Los servicios que brinda esta capa incluyen:

1. Directorio de datos: Establece un catálogo de datos en línea permitiendo la navegación mediante las URI de las bases de datos.
2. Búsqueda de datos: Provee motores de búsqueda de los recursos.
3. Escritorio personalizado: Permite la administración de referencias.
4. Estadísticas y monitoreo de red: Provee estadísticas sobre el acceso a datos y actualizaciones, la valoración de los datos científicos y su visualización.

Finalmente, la capa de aplicación es la que permite la interacción con los usuarios finales según sus necesidades.

b) E2: Europeana

Europeana es la biblioteca digital europea que hace las veces de un portal de búsqueda de colecciones de instituciones culturales de toda Europa. Este portal digital emplea modelos de referencia y vocabularios controlados los cuales pueden ser aplicados dentro del contexto de los datos enlazados. Este caso detalla el impacto y los beneficios de implantar LD en esta biblioteca. Una característica principal de Europeana es que posee un modelo de datos específico Europeana Data Model (EDM) basado en LOD. Por lo tanto, la utilización de LD puede aplicarse para la recolección de datos mejor contextualizados, enriquecer los metadatos ya existentes y proveer un acceso directo a los mismos. Otro impacto relevante es la mejora en los procesos de búsqueda (Ríos-Hilario et al., 2012).

La *Figura 1.6* resume los beneficios del cambio del modelo de Europeana Semantic Elements (ESM) a Europeana Data Model (EDM). Esta última permite la contextualización semántica de los datos por medio de representaciones de objetos que están conectados sistemáticamente a LOD. La adaptación al modelo EDM le permite a Europeana ser compatible con la web semántica y permite la reutilización de ontologías ya existentes. Asimismo, se resuelven los problemas de tener diferentes estándares de metadatos y se genera un enriquecimiento global en todos los niveles de datos, así como en sus modelos de descripción y búsqueda (Ríos-Hilario et al., 2012).



Figura 1.6 Caso Europeana: modelo de datos (Ríos-Hilario et al., 2012)

c) **E3: Open Archives Initiative Protocol for Metadata Harvesting (OIA-PMH)**

OIA-PMH es un protocolo para la recolección de metadatos para lograr la interoperabilidad entre repositorios de librerías digitales. Mediante este protocolo lo que se busca es la integración de repositorios digitales abiertos latinoamericanos, combinando recursos de información existentes en diversas fuentes. Esta iniciativa está aplicada a repositorios de universidades latinoamericanas que usan tecnologías como Eprints y DSpace ya que estas ofrecen la posibilidad de acceder a la información mediante el protocolo OIA-PMH y representar los metadatos mediante el estándar Dublin Core (Piedra et al., 2014).

En OIA-PMH cada repositorio almacena sus recursos digitales de manera individual e independiente. Por otro lado, Dublin Core es el esquema de metadatos para describir recursos digitales. La *Figura 1.7* muestra las ventajas de un marco de publicación de datos de librerías digitales basado en *Linked Data* sobre uno basado solamente en OAI-PMH: en esta última la web solo es una infraestructura de transporte de metadatos, mientras que en *Linked Data* la web posee en sí misma los datos y su semántica (Piedra et al., 2014).

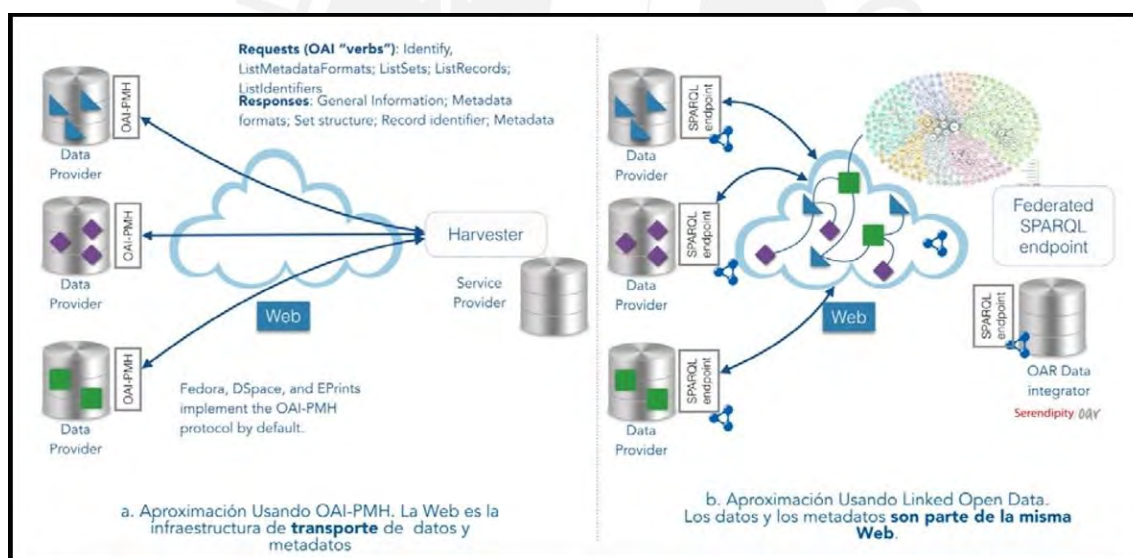


Figura 1.7 Integración de metadatos a través de *Linked Data* (Piedra et al., 2014)

En consecuencia, esta iniciativa propone un marco de trabajo para la extracción de los metadatos usando OIA-PMH para publicarlos como datos enlazados. De acuerdo a la *Figura 1.8* se obtiene el siguiente esquema de trabajo (Piedra et al., 2014):

1. Se seleccionan las fuentes de datos de acuerdo a un determinado proyecto.
2. Se extraen los datos a través del protocolo, se corrigen los datos errados y se transforman para almacenarlos en formato de tripletas OIA.

3. Se realiza un mapeado entre vocabularios y ontologías que permita la interoperabilidad.
4. Se convierten los datos a formato RDF y se limpian los datos para eliminar ambigüedades o cualquier error en el proceso de conversión.
5. Se enlazan los datos por medio de sus relaciones semánticas con fuentes existentes.
6. Se publican y explotan los datos en la web para distintos fines.

Esta iniciativa concluyó con el diseño de dos servicios web, uno para la sugerencia de tópicos y otro para la visualización de datos, los cuales serían integrados en la plataforma Serendipity.

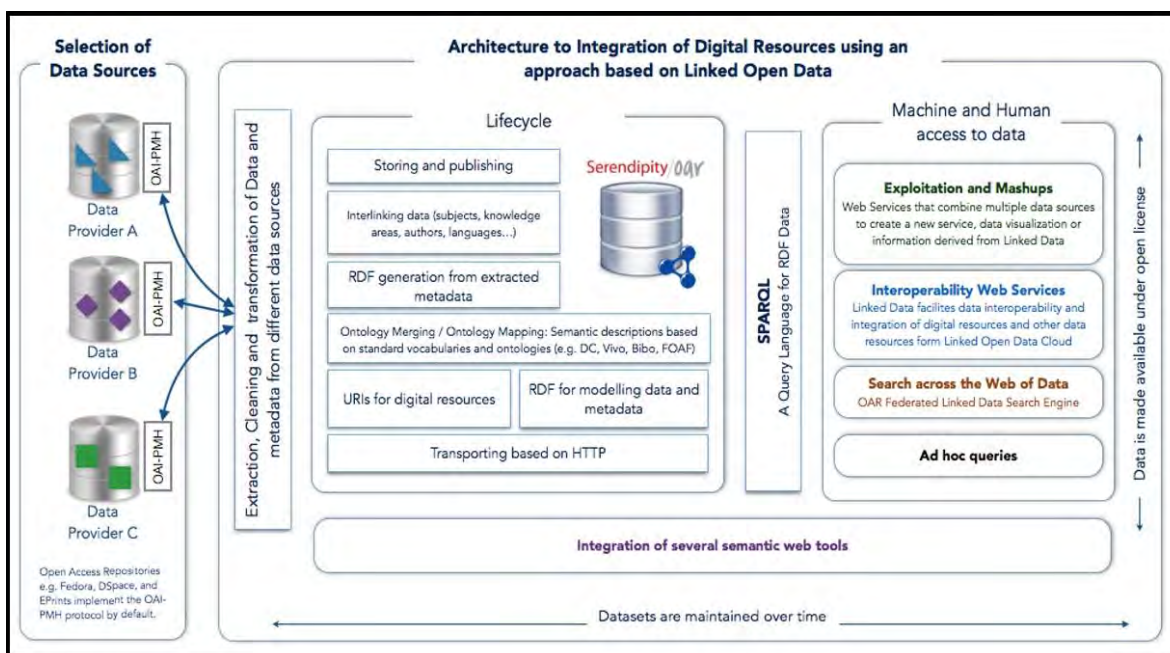


Figura 1.8 Marco de trabajo (Piedra et al., 2014)

d) E4: Open Science

La Universidad de Bielefeld con su centro asociado CITEC (*Center of Excellence Cognitive Interaction Technology*) ha desarrollado una plataforma que permite a los investigadores la administración de sus publicaciones de una forma eficiente mediante *Linked Data* unificando las fuentes propias de la universidad con otras externas como DBPedia. De esta manera, se establecieron tres condiciones básicas para la publicación de los datos (Wiljes et al., 2013):

- Fácil: la publicación de las investigaciones debe constituir un mínimo esfuerzo para los investigadores.
- Útil: La publicación de los datos debe ofrecer un beneficio tanto a la comunidad científica como para el propio investigador.

- Citable: las publicaciones deben ser citables para poder tener contacto con los investigadores.

La *Figura 1.9* muestra la infraestructura necesaria para la publicación y administración de los artículos científicos.

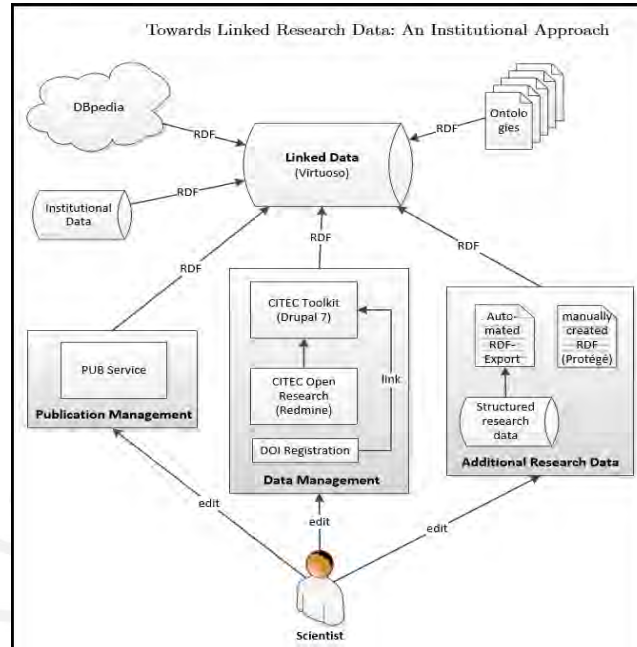


Figura 1.9 Infraestructura Open Science (Wiljes et al., 2013)

De acuerdo con lo que se visualiza se distinguen los siguientes módulos principales (Wiljes et al., 2013):

- Administración de Publicaciones: Engloba un repositorio institucional para el almacenamiento de las publicaciones y permite la visualización de dichas publicaciones para los investigadores que forman parte de la red de Bielefeld.
- Administración de Datos: Consta de tres componentes
 - *CITEC Open Research Platform*: es una aplicación web para alojar proyectos de código abierto. Asimismo, provee a los investigadores un conjunto de funcionalidades para la administración de las publicaciones, notificaciones por email, y la posibilidad de cargar investigaciones digitales en un repositorio central.
 - *Cognitive Interaction Toolkit*: Basado en el sistema de gestión de contenidos Drupal, permite la integración de la información que se aloja en el *CITEC Open Research*. En la interacción de los investigadores con este componente se genera la red de Linked Data.

- Registros DOI: El DOI (*Digital Object Identifier*) es un identificador único para cada dataset, de tal manera que se conozca el origen de la data y pueda ser usada como base en investigaciones posteriores.

1.2.4 Discusión

En base a las preguntas de la revisión, definidas en la etapa de planeamiento, en esta sección se discute los hallazgos basados en las fuentes seleccionadas.

- ¿Cuáles son los problemas o limitaciones del método clásico para estructurar la información en las bibliotecas y repositorios digitales?

Los estudios analizados señalan que uno de los factores principales es el rápido crecimiento de las bases de datos digitales que utilizaron. En el caso de E1, E2, E3 y E4 se presentaron escenarios muy similares, se utilizó una gran cantidad de información digital que estaba distribuida en un conjunto de bases de datos, cada una de las cuales manejaba sus propias estructuras para el manejo de la información. No existía una integración ni un estándar para el manejo de los repositorios digitales y, a medida que la información crecía, su mantenimiento se dificultaba aún más.

- ¿Cuáles fueron las necesidades y objetivos para la búsqueda de una solución al problema?

La necesidad indicada en E1 y E4 era contar con un repositorio centralizado que permitiera una administración integral de las fuentes digitales y que se adapte a su rápido crecimiento. E2 indicaba que la necesidad estaba en la búsqueda centralizada de los metadatos de las publicaciones y así combinar recursos de distintas fuentes o repositorios latinoamericanos. E3 buscaba la unificación de distintas fuentes digitales latinoamericanas de distintos campos de investigación.

- ¿Cuáles fueron las acciones e iniciativas para la resolución del problema?

En los cuatro casos se quería contar con un repositorio centralizado de las investigaciones académicas y científicas. Específicamente E1 y E4 intentaban unificar cada uno de los repositorios digitales que se habían ido creando de manera individual y, a su vez, luego hacer que este repositorio centralizado también se enlace a otros que son de libre acceso. En el

caso de E3 se buscaron instituciones latinoamericanas cuyos repositorios aplicaran las tecnologías de Eprints y DSpace como requisito para que formaran parte del repositorio centralizado, de tal forma el tratamiento de las estructuras de información utilizadas cumplía con un estándar conocido. E2 buscaba ser un repositorio centralizado enlazado a los repositorios digitales de toda Europa, por lo que para cada conjunto de repositorios se implementa un proceso de adaptación de acuerdo a las características de sus estructuras de datos.

- ¿Qué tecnologías se usaron para el planteamiento e implementación de la solución?

En los cuatro casos la tecnología usada fue *Linked Data*. Básicamente, todos los casos estructuran las soluciones en tres componentes o fases principales: la extracción aplicando ontologías, la transformación a RDF y la carga estructurada de la información con la implementación de interfaces para la visualización de la información por parte de los usuarios finales. E3 agregó el uso de OIA-PMH para la recolección de los metadatos exclusivamente, dichos metadatos debían estar estructurados en base a Dublin Core. E4 utilizó *Cognitive Interaction Toolkit* para la integración de la información. E2 utilizó un propio modelo en base a LOD.

- ¿Cuáles fueron los inconvenientes y desafíos para ejecutar la solución planteada?

E1 indica que un principal inconveniente era la diversidad de estructuras utilizadas y, debido a la gran cantidad de información, el proceso de extracción y transformación era susceptible a errores o ambigüedades. E4 fue una iniciativa que buscaba que los investigadores empezaran a ingresar la información, el desafío consistía en construir una plataforma que permitiera la facilidad de uso para los investigadores y que el repositorio ofreciera beneficios para la red de los investigadores. E3 también indicaba la complejidad de la extracción de los metadatos y en su solución se contemplaba una fase para la corrección de datos ambiguos. Finalmente, E2 contemplaba no reemplazar la información de los repositorios en base a la nueva estructura basada en LD, sino que buscaba su extracción para integrarlo en un nuevo repositorio en el cual se pueda agregar más información y, a su vez, enlazarlo con cada uno de los repositorios de origen. La complejidad radica en los formatos utilizados en los repositorios, algunos de los cuales pueden tener más de 20 años y han pasado por múltiples fases de actualización y la información en sí misma no es íntegra.

1.2.5 Conclusiones del Estado del Arte

A través de la revisión realizada se encontraron soluciones mediante el uso de *Linked Data* utilizando un marco de trabajo similar en cada una de ellas, el cual implica la utilización de ontologías, la transformación de los datos al formato RDF y finalmente el uso de una interfaz para que el usuario final pueda hacer uso de los datos de las publicaciones. En las soluciones analizadas se opta por hacer uso de las ontologías propias de los *datasets*. Para el presente proyecto se hará uso de una ontología dinámica adaptada al dominio de las investigaciones académicas y científicas para la extracción de datos desde distintos conjuntos de datos.

1.3 Marco Conceptual

En esta sección se presentan los conceptos relevantes para la comprensión del presente proyecto, el cual consiste en la implementación de una herramienta de software que permita la búsqueda de información de publicaciones académicas en distintas bases de datos estructuradas bajo el método de los Datos Enlazados (LD). En tal sentido, se expondrán los conceptos más importantes respecto a *Linked Data* y cómo las necesidades actuales en el manejo de la información han permitido su crecimiento y rápida evolución.

1.3.1 Ontología

Una ontología es una estructura o definición formal que modela el conocimiento dentro de un dominio o contexto dado, especificando de manera explícita un conjunto de conceptos y las relaciones entre ellos (Gruber, 1995). Es decir, entendiéndose que una conceptualización es una abstracción que permite la representación de objetos o entidades presentes en el mundo, las ontologías permiten la descripción o formulación de la abstracción por medio de la definición formal de los objetos, de sus conceptos y de las relaciones que se dan entre ellos (Mata, Crespo, & Maña, 2011).

1.3.2 *Linked Data*

Linked Data es la forma en la cual se vinculan los datos que se encuentran publicados y distribuidos en la Web. De esta manera, a partir de los datos que están publicados en la Web se puede hacer referencia a otros datos de la misma forma en que las páginas web se vinculan unas con otras mediante enlaces o hipervínculos (W3C, s. f.). Este cúmulo de información ha conducido a la creación de un espacio de datos globales conocida como la Web de los Datos (Web Data). La

posibilidad de vincular los datos provenientes de distintas fuentes de información abre horizontes prometedores en cuanto a las aplicaciones prácticas en distintas áreas de investigación y desarrollos tecnológicos.

Entre las aplicaciones prácticas se puede mencionar a desarrollos en el uso de LD en las bibliotecas digitales académicas orientados al apoyo de la investigación científica que permiten la relación de significados entre datos e información (Ávila Barrientos, 2016); por lo tanto, “tiene un impacto en áreas como la recuperación de la información y el acceso a los recursos de información digital” (Ávila Barrientos, 2016). Otra área que aprovecha la tecnología de LD es Inteligencia Artificial para la construcción de redes de datos estructurados para el entendimiento de las máquinas, es decir que pueden ser procesados de forma automatizada. Otras áreas pueden estar más enfocadas al consumo masivo como es el caso del comercio electrónico que, haciendo uso de LD en los catálogos en línea, permiten al usuario una búsqueda más amplia y de mayor exactitud. En términos concretos, LD puede ser utilizada en distintas áreas del conocimiento según las necesidades en el manejo de la información.

Desde el comienzo del Internet, con la *World Wide Web*, los paradigmas de intercambio de conocimiento han cambiado radicalmente, eliminando las barreras para acceder y compartir información de todo tipo: todo ha pasado a formar parte de un espacio de información global (Méndez Rodríguez, 2001). Sin embargo, aún existen límites en cuanto al orden de esta misma información y de la proliferación de datos que pueden resultar equivocados o poco útiles para los propósitos por la cual se realiza la búsqueda (Peis et al., 2003). Estos inconvenientes se deben a que en la actualidad se ha hecho muy fácil y casi universal la posibilidad de publicar información, encontrándola esparcida sin un estándar común en sus estructuras de documentos (Méndez Rodríguez, 2001).

El método actual para encontrar información es a través de los Navegadores Web que permiten realizar consultas de información por medio de sus motores de búsqueda, los cuales cuentan con algoritmos complejos para que dichos resultados sean potencialmente relevantes para el usuario. Sin embargo, los datos no han sido tratados como tal, sino que han estado habilitados por medio de formatos como CSV, XML, HTML, etc. En tal sentido, se han sacrificado aspectos importantes como la estructura y la semántica de los datos presentados, por lo cual no se pueden hacer relaciones entre estos conjuntos de datos (Bizer, Heath, & Berners-Lee, 2009). La forma actual de funcionamiento de la web posee principalmente la limitación de presentar documentos HTML que contienen información estructurada en lenguaje natural que es entendible para las personas, pero no para las propias máquinas. Las búsquedas mediante los navegadores son imprecisas para los fines de los usuarios y se pueden encontrar enlaces entre documentos sin un

significado concreto que son de poca utilidad para la búsqueda exacta de información relevante (“Guía Breve de Web Semántica”, s. f.) .

Por ejemplo, en el caso de las publicaciones académicas es difícil encontrar artículos académicos relevantes para investigaciones que amplíen las mismas o sirvan de base para explorar nuevos horizontes (Méndez Rodríguez, 2001). Esta dificultad radica en que un buscador común no es capaz de realizar búsquedas eficientes en cuanto al contenido que puede presentar este tipo de publicaciones: no existe la forma de contextualizar la búsqueda a aspectos como datos relevantes y palabras clave ya sea en los títulos, la bibliografía o en el propio contenido (Peis et al., 2003). Los motores de búsqueda están limitados a encontrar documentos y no datos dentro de un contexto específico. Se puede afirmar que al utilizar un motor de búsqueda el usuario insertará una cadena de texto, el buscador recuperará la mayor cantidad de similitudes que pueda haber y las mostrará como resultado. Dichos documentos (páginas web) pueden estar dentro del contexto que se desea o simplemente estar completamente fuera del área que es el objetivo de la investigación (“Guía Breve de Web Semántica”, s. f.).

Debido a estas limitaciones se ha necesitado extender la web para alcanzar un espacio de información global donde los documentos y los datos estén enlazados, la cual fue la idea inicial de Tim Berners Lee (Bizer et al., 2009). *Linked Data* consiste en el uso de la web para construir vínculos entre datos de fuentes distintas o pertenecientes a la misma. Básicamente se trata de datos que están publicados en la Web cuyos significados están explícitamente definidos, poseen vínculos a otros datos y que también pueden ser vinculados desde otros conjuntos de datos externos (Peset, Ferrer, & Subirats, 2011). La web de los datos también es llamada la web semántica porque precisamente los datos vinculados están dotados de mayor significado y, por tanto, de mayor semántica mediante el uso de una estructura común (“Guía Breve de Web Semántica”, s. f.). De acuerdo a la *Figura 1.10* se visualiza cómo es que la web común, que solo era capaz de vincular documentos, fue extendida para realizar una vinculación de los datos, el cual permite la extracción de información con un nivel de granularidad mucho mayor.

Existen similitudes en cuanto a las tecnologías y conceptos usados en la Web común (Web del hipertexto) y la Web de los datos. Mientras que en la primera la unidad primaria está basada en documentos HTML conectados por hipervínculos no tipificados, *Linked Data* se basa en documentos que contienen datos en formato RDF (Peset, Ferrer, & Subirats, 2011). Sin embargo, una gran diferencia es que el uso de RDF tiene el propósito de no sólo vincular estos documentos sino de generar estructuras de datos que vinculan cosas arbitrariamente (W3C, s. f.). El resultado de este mecanismo de vinculación de los datos es lo que se mencionó previamente como la Web de los datos, el cual se describe como la Web de las cosas existentes en el mundo que está descrita por

datos en la Web. La Web de los datos permite la abstracción de las cosas existentes en el mundo, de tal manera que estén accesibles y vinculados entre sí sin la necesidad de que tengan una relación directa o que pertenezcan a un mismo conjunto de datos.

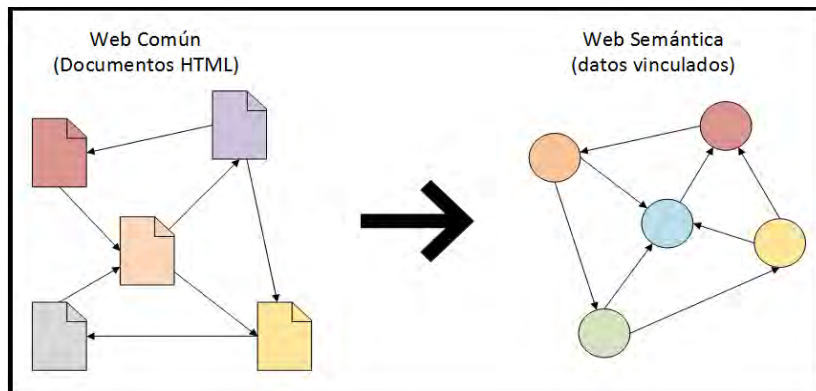


Figura 1.10 Representación de la web común y la web de los datos. Elaboración propia.

Linked Data se rige de las tecnologías usadas en la Web, pero aplicando su propia arquitectura y estándares. Esta tecnología está basada en dos pilares: Uniform Resource Identifiers (URI) y HiperText Transfer Protocol (HTTP⁴). Asimismo, existen cuatro principios básicos, definidos en el 2006 por Berners Lee, para la publicación de datos en la Web de tal forma que todo lo que se publique llegue a formar parte de un simple espacio de datos global (W3C, s. f.):

- a) El uso de URI (Uniform Resource Identifier) para identificar recursos.
- b) Uso de URI HTTP (Hiper Text Transfer Protocol) para la localización de los recursos.
- c) Proveer información útil al buscar una URI por medio del estándar RDF.
- d) Incluir vínculos a otras URI.

Estos principios permiten el uso de estándares para la representación y acceso a los datos en la web y, por otro lado, propaga el uso de hipervínculos entre datos de distintas fuentes (“LinkedData - W3C Wiki”, s. f.). Los conceptos involucrados en estos cuatro principios son:

a) *Uso de URI*

URI⁵ es una cadena de caracteres que identifica de manera única y sin ambigüedades a cualquier recurso presente en la red o la Web de datos. Esta cadena cuenta con la siguiente estructura (Berners-Lee, s. f.): **Esquema://Autoridad/Ruta/Consulta#Fragmento.**

⁴ <https://www.w3.org/Protocols/>

⁵ <https://www.w3.org/wiki/URI>

- Esquema: Es una especificación para asignar identificadores. El esquema http es el más frecuente encontrado en Internet, así también existen otros como ftp para transferencias de archivos y ssl para páginas seguras de Internet. Los tres mencionados son, a su vez, protocolos de comunicación o de acceso a los recursos, pero también existen otros como tag y cid.
- Autoridad: Define la autoridad de nombre de donde proviene el recurso.
- Ruta: Identifica el recurso dentro de un ámbito específico dado por el esquema y la autoridad.
- Consulta: Identifica también un recurso -al igual que la ruta-, pero se presenta en un par “clave=valor” y comienza con el carácter ‘?’. No es un componente obligatorio.
- Fragmento: Define solo una parte o sección del recurso principal y comienza con un ‘#’ para identificar a este componente. Al igual que el componente de consulta, este tampoco es obligatorio.

El ejemplo práctico se basará en una consulta a la Semantic Web Journal (SWJ), la cual es un dataset de investigadores, publicaciones, y categorías de artículos (Pascal Hitzler & Krzysztof Janowicz, s. f.). En la *Figura 1.11* se especifica la URI de una entidad que es un artículo académico:



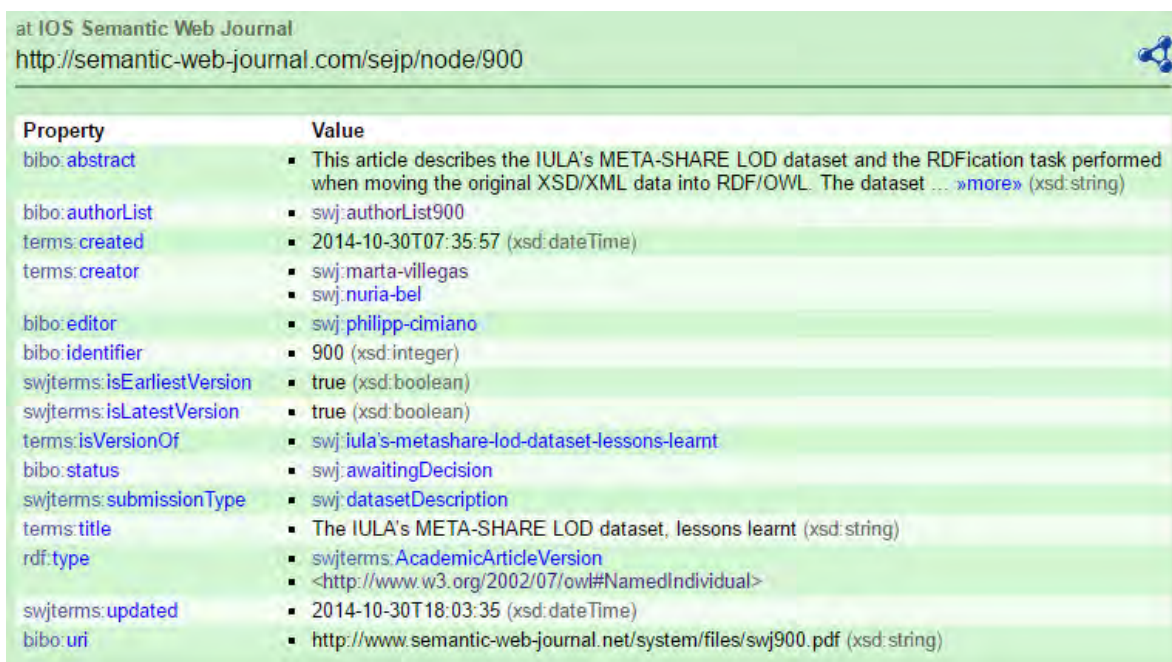
Figura 1.11 Ejemplo de URI

Mediante el uso de URI es posible realizar la identificación inequívoca de un recurso específico que se aloja en la Web. Si bien la URL (*Uniform Resource Locators*) es el estándar conocido para el direccionamiento de documentos y otras entidades presentes en la Web, las URI proveen un significado más genérico para identificar cualquier tipo de entidad que exista en el mundo.

b) Uso de URI HTTP

HTTP es el protocolo orientado a transacciones utilizado en la Web. Se basa en la interacción de petición-respuesta. Cuando se requiere acceder a una entidad esto se realiza mediante su URI que es desreferenciada por medio del protocolo HTTP (Berners-Lee, 2006). En este sentido, HTTP provee un mecanismo simple pero universal para la recuperación de recursos que pueden ser

serializados como una cadena de bytes, o para la recuperación de descripciones de entidades que por sí mismos no pueden ser enviados o distribuidos por la red (Berners-Lee, 2006). De acuerdo al ejemplo de la *Figura 1.12*, cuyo esquema es precisamente http, se logra la desreferenciación del recurso que identifica la URI. Mediante esta URI se accede al recurso identificado, esta forma de identificación es única y no permite ambigüedades. La información que se recupera puede ser visualizada en distintos tipos de estructuras como el “turtle document” o RDF/XML; a continuación, se visualiza la primera forma, el cual es una representación textual de un grafo RDF (“RDF 1.1 Turtle”, s. f.):



Property	Value
bibo:abstract	▪ This article describes the IULA's META-SHARE LOD dataset and the RDFication task performed when moving the original XSD/XML data into RDF/OWL. The dataset ... »more» (xsd:string)
bibo:authorList	▪ swj:authorList900
terms:created	▪ 2014-10-30T07:35:57 (xsd:dateTime)
terms:creator	▪ swj:marta-villegas ▪ swj:nuria-bel
bibo:editor	▪ swj:phillip-cimiano
bibo:identifier	▪ 900 (xsd:integer)
swjterms:isEarliestVersion	▪ true (xsd:boolean)
swjterms:isLatestVersion	▪ true (xsd:boolean)
terms:isVersionOf	▪ swj:iula's-metashare-lod-dataset-lessons-learnt
bibo:status	▪ swj:awaitingDecision
swjterms:submissionType	▪ swj:datasetDescription
terms:title	▪ The IULA's META-SHARE LOD dataset, lessons learnt (xsd:string)
rdf:type	▪ swjterms:AcademicArticleVersion ▪ <http://www.w3.org/2002/07/owl#NamedIndividual>
swjterms:updated	▪ 2014-10-30T18:03:35 (xsd:dateTime)
bibo:uri	▪ http://www.semantic-web-journal.net/system/files/swj900.pdf (xsd:string)

Figura 1.12 Recurso recuperado mediante URI (“IOS Semantic Web Journal”, s. f.)

c) *Resource Description Framework (RDF⁶)*

Mientras HTML permite la vinculación de documentos en la Web, RDF provee un modelo genérico de datos basado en grafos cuya estructura y datos vinculados permiten describir cosas presentes en el mundo (abstracción) (Méndez Rodríguez, 2001). Es decir, RDF es un modelo de datos que permite especificar o describir las relaciones entre los recursos web (Ríos-Hilario et al., 2012). El modelo se basa en la terna sujeto-predicado-objeto y de esta terna, el sujeto y el objeto son dos URI que identifican un recurso, o también una URI y una cadena de caracteres explícita. El

⁶ <https://www.w3.org/RDF/>

predicado especifica cómo están relacionados el sujeto y el objeto, y también es representado por un URI.

De acuerdo a la *Figura 1.12* la URI “<http://semantic-web-journal.com/sejp/node/900>” que ha sido desreferenciada es el sujeto y cada uno de los registros que contienen los campos de propiedad y valor conforman la terna de sujeto-predicado-objeto. Para fines demostrativos el sujeto será nombrado por su ID (bibo: identifier) 900; por lo tanto, se tiene la siguiente estructura dada por la *Figura 1.13*:

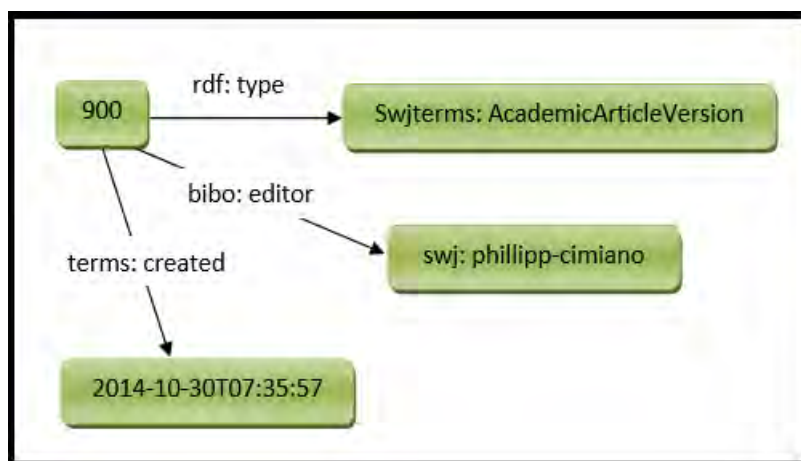


Figura 1.13 Ejemplo de sujeto-predicado-objeto. Elaboración propia.

Como se aprecia, el predicado es la propiedad, atributo o relación entre el sujeto y el objeto, mientras el objeto es el valor. Se escogieron tres distintos predicados: “rdf: type”, “bibo: editor” y “terms: created”; para los dos primeros el objeto (valor) también son dos URI, mientras el tercero es una cadena de texto. Las especificaciones RDF definen un vocabulario como base para el esquema RDF, dentro de esta se declaran clases y propiedades. El esquema RDF es un conjunto de clases con ciertas propiedades utilizando el modelo de datos RDF con la intención de estructurar recursos RDF, así como la formación de vocabulario RDF y elementos básicos para la descripción de ontologías (W3C, 2014). Asimismo, existen formatos de serialización como RDF/XML, Turtle, N-Triples, etc. La serialización radica su importancia en la posibilidad de la transmisión de esta información como una serie de bytes o en formatos más entendibles para el razonamiento humano (W3C, 2014).

d) Incluir enlaces a otras URI

Al incluir enlaces a otras URI los datos de un conjunto específico quedarán enlazados con otras fuentes de información brindando valor agregado a la información que ya se muestra. De esta

manera se logra que los datos no queden aislados, que la información acreciente su valor ya que se puede acceder a fuentes más especializadas para ciertos datos que simplemente se muestran pero que no son parte esencial del conjunto. Del mismo modo, otras fuentes externas pueden hacer uso y vincular a los datos propios. Esta dinámica es la misma que se realiza con los enlaces HTML, en el cual a partir de un documento se puede acceder a otro mediante un vínculo dentro de la información que se presenta. La importancia de esta conectividad de los datos es la utilidad para que cualquier recurso sea enriquecido por más información especializada, además de que no existe la necesidad de que a partir de un recurso se tenga que llegar a otro del mismo conjunto, sino que pueden no tener relación directa (W3C, s. f.).

1.3.3 La Web clásica y la Web de los datos

Con el uso de HTTP como el mecanismo de recuperación de recursos y el RDF como el modelo de datos para representar dichos recursos se logra la formación de la Web de los datos, la cual contiene una estructura muy semejante a la Web clásica de documentos HTML y, por tanto, posee las mismas propiedades (Bizer et al., 2009). La Web de los datos es genérica y puede contener cualquier tipo de dato y no existe restricción alguna para su publicación, del mismo modo que se da en la Web clásica al momento de publicar algún documento HTML. Esto también ha permitido su rápido crecimiento, tal como sucedió con la *World Wide Web* (Bizer et al., 2009).

Asimismo, cualquier persona puede realizar esta publicación de datos y no están restringidos en la elección de vocabularios específicos para representar la data. La libertad de publicación y parametrización de los datos permite un gran potencial de crecimiento acelerado en la formación de este conjunto global de datos, cuya importancia radica en la capacidad de que un conjunto de datos, en un contexto específico, puede ser retroalimentado o repotenciado por conjuntos externos vinculados por medio de las URI. En ese sentido, estas entidades se conectan por los vínculos RDF componiendo un grafo de data global que expande las fuentes de los datos y que permite el descubrimiento de nuevas fuentes (Papadakis, Kyprianos, & Stefanidakis, 2015).

Así como existen estándares y arquitecturas para el desarrollo de aplicaciones que puedan ser usadas en la Web clásica, la Web de los datos también brinda una perspectiva estandarizada para cualquier desarrollo de aplicaciones. La data está estrictamente separada de cualquier formato y representación y debe ser capaz de describirse a sí misma. En tal sentido, si una aplicación que consume Linked Data encuentra data que posee vocabulario no definido, el método será desreferenciar las URI necesarias que contienen dicho vocabulario para determinar su definición (Bizer et al., 2009).

Una aplicación de este tipo es comparable a una Web API que contiene modelos de datos heterogéneos e interfaces de acceso. Estas aplicaciones no son implementadas en base a un conjunto de fuentes de datos predeterminados, sino que puede ir creciendo con nuevas fuentes debido a la presencia de vínculos RDF (Guha et al., 2016). En tiempo de ejecución la aplicación será capaz de mostrar conjuntos de datos que permitirán vínculos a otras fuentes, uno de los principios de Linked Data. Estas implementaciones son posibles debido a la existencia de la Web de los datos, la cual es libre y está abierta a cualquier tipo de uso de toda la data global existente (Guha et al., 2016).

En tal sentido, existen proyectos como Linking Open Data, el cual tiene como objetivo el extender y publicar distintos conjuntos de datos libres en formato RDF en la Web (Bizer et al., 2009). También existen proyectos promovidos por la Unión Europea como LATC⁷ (*linked open data around the clock*), PlanetData⁸, DaPaas⁹ (*Data and Platform as a Service*) y Linked Open Data 2 (“The Linking Open Data cloud diagram”, s. f.). Incluso, entendiendo la importancia de la vinculación y administración de los datos, existe un portal de datos abiertos administrado por la Unión Europea creado en el 2012. Este portal permite poner a disposición de las personas la información de las instituciones académicas pertenecientes a la Unión Europea (“Acerca de | Portal de datos abiertos”, s. f.).

1.3.4 Metadatos

Ya se mencionó que RDF es un modelo de datos para metadatos¹⁰; en ese sentido, los metadatos son relevantes ya que proveen información extra sobre los datos publicados. Los metadatos sirven para conocer la fuente de los datos y otras características propias que permiten reconocer la calidad y otros aspectos que son importantes para que una persona que accede a la data la califique como lo necesario de aquello que busca. Los metadatos son de gran importancia para la generación de información descriptiva relevante sobre los propios datos (“Resource Description Framework (RDF) Model and Syntax Specification”, s. f.).

En el ejemplo de la *Figura 1.13*, previamente explicada, los datos que se recuperan por medio de la URI constituyen los metadatos del artículo. Estos metadatos se pueden extraer en formato XML y presenta la misma información que el formato “Turtle”; los metadatos brindan información relevante que no estará presente dentro del contenido tales como tipo de artículo, editor, autores, versión, el vínculo a la última versión, el estado, entre otros datos relevantes. Parte de estos metadatos se presentan en la *Figura 1.14*:

⁷ <http://aksw.org/Projects/LATC.html>

⁸ <https://planet-data.org/>

⁹ <http://project.dapaas.eu/>

¹⁰ <https://www.w3.org/Metadata/Activity.html>

```

<?xml version="1.0" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://purl.org/ontology/hibo/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:j.1="http://purl.org/dc/terms/"
  xmlns:j.2="http://semantic-web-journal.com/ontology#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <j.2:AcademicArticleVersion rdf:about="http://semantic-web-journal.com/sejp/node/900">
    <j.0:status rdf:resource="http://semantic-web-journal.com/sejp/awaitingDecision"/>
    <j.0:editor rdf:resource="http://semantic-web-journal.com/sejp/philipp-cimiano"/>
    <j.1:creator rdf:resource="http://semantic-web-journal.com/sejp/nuria-bel"/>
    <j.1:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The IULA's META-SHARE LOD dataset, lessons learnt</j.1:title>
    <j.1:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
    >2014-10-30T07:35:57</j.1:created>
    <j.0:identifier rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
    >900</j.0:identifier>
    <j.2:submissionType rdf:resource="http://semantic-web-journal.com/sejp/datasetDescri
    <j.2:isEarliestVersion rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</j.2:isEarliestVersion>
    <j.0:abstract rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >This article describes the IULA's META-SHARE LOD dataset and the RDFication task pe
    <j.1:creator rdf:resource="http://semantic-web-journal.com/sejp/marta-villegas"/>

```

Figura 1.14 Metadatos en formato XML (“IOS Semantic Web Journal”, s. f.)

1.3.5 Datasets

Los datasets son los grandes conjuntos de datos que se han ido creando. Son las grandes bases de datos con contenido de Linked Data. Estos conjuntos son los que se vinculan tanto internamente como externamente con otros datasets. Los datasets conforman lo que se mencionó anteriormente como el espacio de data global -Web de los datos- (Bizer et al., 2009). Entre los datasets más importantes se encuentran los siguientes:

- Wikidata¹¹: Es un proyecto que en la actualidad cuenta con aproximadamente 16 millones de ítems de data vinculada. Es libre y es un proyecto en paralelo al de Wikimedia (Wikipedia, Wikivoyage, etc).
- DBPedia¹²: Es un proyecto que consiste en la extracción de información de Wikipedia. Cuando se publica un documento en esta Web, al mismo tiempo se generan datos RDF que estructuran la información y la vinculan con el resto de los conjuntos de datos. DBPedia tiene conexión con bases de datos de otros proyectos como GeoNames¹³.
- GeoNames: Es una base de datos geográfica cuya información se puede acceder mediante sus URI, teniendo la información como una página HTML o la descripción de los recursos en RDF.

¹¹ https://www.wikidata.org/wiki/Wikidata:Main_Page

¹² <http://wiki.dbpedia.org/>

¹³ <http://www.geonames.org/>

Todos los datasets existentes no son de libre acceso: algunos datasets no permiten el acceso gratuito, otros se pueden consultar, pero solo es posible obtener los metadatos (grafos RDF) mas no los contenidos o textos completos debido a las licencias y derechos de autor. Un proyecto ambicioso que aplica los principios de LD es el proyecto de Linking Open Data (LOD¹⁴), el cual tiene el fin de interconectar todos los datasets disponibles bajo licencias abiertas. Todos estos datasets conforman la nube de Linked Open Data. Desde la creación de este proyecto en el 2007 la nube ha estado en constante crecimiento según se muestra en la *Figura 1.15*. Hasta abril 2018 la nube ya está conformada por 1,184 Datasets. Asimismo, la *Figura 1.16* muestra los conjuntos de datos interconectados, cada color representa un conjunto de áreas del conocimiento asociados.

	White	Colored	Graph file	Dataset list	Datasets
2018-04-30		png	svg	json	1,184
2017-08-22		png	svg	json tsv	1,163
2017-02-20		png	svg		1,139
2017-01-26		png	svg		1,146
2014-08-30	png pdf	svg	png pdf	svg	570
2011-09-19	png pdf	svg	png pdf	svg	295
2010-09-22	png pdf	svg	png pdf	svg	203
2009-07-14	png pdf	svg			95
2009-03-27	png pdf	svg	png pdf	svg	93
2009-03-05	png pdf	svg	png pdf	svg	89
2008-09-18	png pdf	svg			45
2008-03-31	png pdf	svg			34
2008-02-28	png pdf	svg			32
2007-11-10	png pdf	svg			28
2007-11-07	png				28
2007-10-08	png				25
2007-05-01	png				12

Figura 1.15 Crecimiento de Linked Open Data (“The Linking Open Data cloud diagram”, s. f.)

¹⁴ <https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

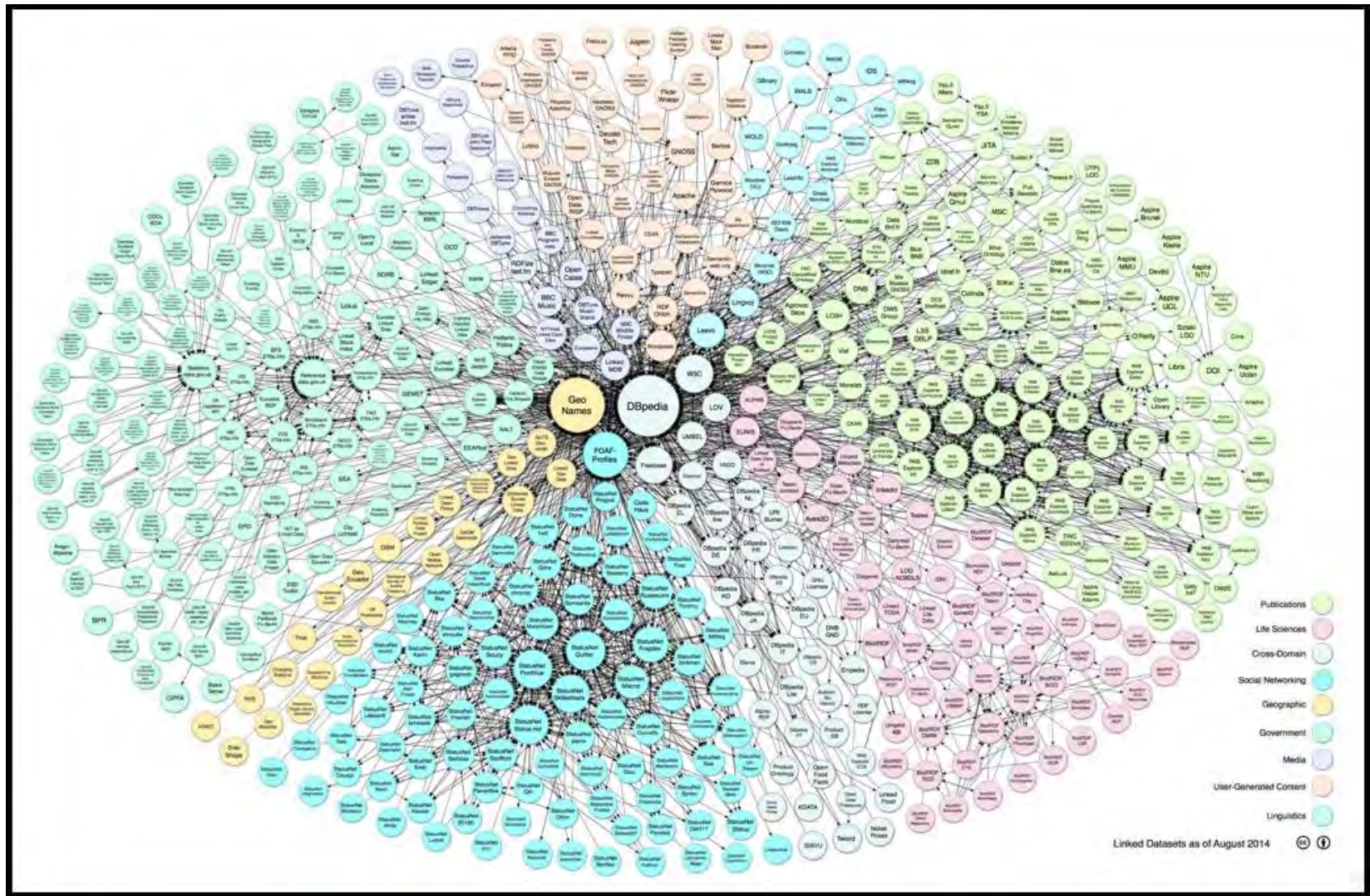


Figura 1.16 Representación de distintos conjuntos de datos vinculados que conforman Linked Open Data (“The Linking Open Data cloud diagram”, s. f.)

CAPÍTULO 2: Objetivos y Resultados Esperados

En el presente capítulo se describen los objetivos planteados para el presente proyecto, así como los resultados esperados relacionados a cada uno de los objetivos.

2.1 Objetivo General

El objetivo del presente proyecto es la implementación de un software para la búsqueda de publicaciones académicas y científicas peruanas presentes en bases de datos estructuradas bajo el método de los Datos Enlazados y que son de libre acceso.

2.2 Objetivos Específicos

1. OE1: Diseñar e implementar una ontología dinámica que permita la descripción de investigadores y académicos peruanos en base a las ontologías de distintos conjuntos de datos existentes (*datasets*).
2. OE2: Diseñar e implementar un método de extracción de los datos presentes en la nube de LOD.
3. OE3: Diseñar e implementar un método de transformación de los datos al formato RDF/XML y Turtle.
4. OE4: Diseñar e implementar una interfaz web para la búsqueda y visualización de la información de las publicaciones académicas y científicas peruanas.

2.3 Resultados Esperados

A continuación, se establecen los resultados esperados en base a cada objetivo específico planteado:

1. Resultados Esperados del OE1
 - RE1.1: La ontología que permita describir el contexto de los autores peruanos y sus publicaciones. Será una ontología adaptativa de acuerdo a los *datasets* en los que se podrá realizar la búsqueda.
 - RE1.2: Método de verificación de la ontología.
2. Resultados Esperados del OE2
 - RE2.1: Funciones bajo el lenguaje de SPARQL para la extracción de la información de los *datasets*.
3. Resultados Esperados del OE3

RE3.1: Funciones de transformación de metadatos estructurados bajo el esquema de RDF/XML y Turtle.

4. Resultados Esperados del OE4

RE4.1: Módulo de búsqueda de publicaciones

RE4.2: Módulo estadístico de publicaciones y autores

2.4 Herramientas, Métodos y Metodologías

En esta sección se presentan las herramientas, métodos y metodologías a ser utilizadas para la consecución de cada uno de los resultados esperados definidos previamente. A nivel transversal, para la implementación de toda la solución, se cuenta con el lenguaje de programación Java bajo el *framework* de Spring y el entorno de desarrollo integrado Eclipse.

Resultados Esperados	Herramientas o Métodos
<p>RE1.1: La ontología que permita describir el contexto de los autores peruanos y sus publicaciones. Será una ontología adaptativa de acuerdo a los <i>datasets</i> en los que se podrá realizar la búsqueda.</p> <p>RE1.2: Método de verificación de la ontología</p>	<ul style="list-style-type: none"> • Herramientas: <ul style="list-style-type: none"> ➤ Framework Jena ➤ Protegé ➤ Ontology Web Language OWL • Metodología: <ul style="list-style-type: none"> ➤ Método de preguntas de verificación (Competency Questions)
<p>RE2.1: Funciones bajo el lenguaje de SPARQL para la extracción de la información de los <i>datasets</i>.</p>	<ul style="list-style-type: none"> • Herramientas: <ul style="list-style-type: none"> ➤ SPARQL <i>Query Language</i> ➤ <i>Resource Description Framework</i> RDF
<p>RE3.1: Funciones de transformación de metadatos estructurados bajo el esquema de RDF/XML y Turtle.</p>	<ul style="list-style-type: none"> • Herramientas: <ul style="list-style-type: none"> ➤ Framework Jena ➤ Base de Datos MySQL
<p>RE4.1: Módulo de búsqueda de publicaciones</p>	<ul style="list-style-type: none"> • Herramientas <ul style="list-style-type: none"> ➤ Lenguaje de programación JAVA ➤ <i>Framework</i> Spring

	<ul style="list-style-type: none"> ➤ <i>Base de Datos MySQL</i> ➤ <i>SPARQL Query Language</i> ➤ Herramienta de modelamiento StarUML • Metodologías: <ul style="list-style-type: none"> ➤ RUP ➤ UML (<i>Unified Modeling Language</i>)
RE4.2: Módulo estadístico de publicaciones y autores	<ul style="list-style-type: none"> • Herramientas <ul style="list-style-type: none"> ➤ Lenguaje de programación JAVA ➤ <i>Framework Spring</i> ➤ <i>Base de Datos MySQL</i> ➤ <i>SPARQL Query Language</i> ➤ Herramienta de modelamiento StarUML • Metodologías: <ul style="list-style-type: none"> ➤ RUP ➤ UML (<i>Unified Modeling Language</i>)

2.4.1 Herramientas

Lenguaje de programación Java

Java es un lenguaje de programación orientado a objetos y multipropósito. Asimismo, los programas desarrollados en Java son multiplataforma ya que pueden ser ejecutados en distintos sistemas operativos debido al uso de Java Virtual Machine JVM. Actualmente, Java es utilizado en la implementación de aplicaciones empresariales cliente-servidor de escritorio, así como tecnología web (Deitel & Deitel, 2015).

Spring

Es un *framework* para el desarrollo de aplicaciones en Java y es de código abierto. Actualmente, en su versión 5.0 es uno de los *frameworks* más estables y robustos para Java. Dado que su primera versión fue lanzada en el 2002, ya tiene el nivel de madurez para el desarrollo de aplicaciones estables y que puedan ser mantenidas a largo plazo con el bajo riesgo

de que el *framework* sea discontinuado o existan problemas en la migración a una nueva versión. Así mismo, Spring cuenta con amplia documentación¹⁵ a disposición para el entendimiento de cada uno de sus componentes y módulos (Johnson et al., 2016). Para propósitos del proyecto, Spring permitirá la implementación del software mediante la arquitectura Modelo-Vista-Controlador (MVC).

Jena

Jena¹⁶ es un *framework* de Java para el desarrollo de aplicaciones con *Linked Data*. Incluye una API para el modelamiento de ontologías, así como el uso de RDF y SPARQL. Asimismo, también contiene un motor de inferencia y un motor de búsqueda. Jena será usado para programar bajo las ontologías de RDF y OWL.

Protégé

Protégé¹⁷ es un editor de ontologías de código abierto y un marco para la construcción de sistemas inteligentes. Está basado en Java y fue desarrollado por la Escuela de Medicina de la Universidad de Stanford dentro del Centro de Investigación de Informática Biomédica (“protégé”, s. f.). Servirá como software de apoyo para la creación de la ontología dinámica.

Lenguaje de Consultas SPARQL (Protocol and RDF Query Language)

SPARQL es un lenguaje estandarizado para la realización de consultas de grafos RDF. Esta tecnología es muy importante para el desarrollo de Webs Semánticas. Es equivalente al SQL, por lo cual existe el lenguaje de consulta y el motor para el almacenamiento y recuperación de los datos, conocido como SPARQL Endpoint. Asimismo, también existen distintas implementaciones que hacen uso de este estándar. Los resultados de una consulta en SPARQL pueden ser devueltos en formatos comúnmente conocidos XML, JSON, CSV (valores separados por comas) o TSV (valores separados por tabulaciones) (W3C, 2013).

De acuerdo a la *Figura 2.1* se tiene la estructura básica de una consulta en SPARQL. La palabra clave PREFIX permite asociar una etiqueta a una URI (Valencia Castillo, s. f.); por ejemplo, la etiqueta dcterms está asociado a “<http://purl.org/dc/terms/>”. La consulta, al igual que en SQL, comienza con la palabra clave SELECT para determinar los campos o datos que se devolverán como resultado. La palabra WHERE especifica las condiciones que se deben

¹⁵ <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/>

¹⁶ <https://jena.apache.org/index.html>

¹⁷ <http://protege.stanford.edu/>

cumplir para que los datos sean recuperados. Esta consulta arrojará como resultado un listado de músicos españoles con los datos de sus nombres, fechas de nacimiento y fechas de fallecimiento.

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbp: <http://dbpedia.org/ontology/>
SELECT ?musico ?nombreMusico ?fechaNacimiento ?fechaFallecimiento
WHERE {
  ?musico dcterms:subject
  <http://dbpedia.org/resource/Category:Spanish_musicians>;
  rdfs:label ?nombreMusico ;
  dbp:birthDate ?fechaNacimiento ;
  dbp:deathDate ?fechaFallecimiento .
  FILTER (LANG(?nombreMusico) = "es")
}
```

Figura 2.1 Ejemplo de consulta en SPARQL (“SPARQL”, 2015)

Ontology Web Language OWL

Es un lenguaje para la web semántica que permite el procesamiento de contenido de información. OWL añade mayor vocabulario que el definido por el esquema de RDF, permitiendo extender las ontologías con un mayor contenido semántico y de relaciones entre los recursos RDF (“OWL Web Ontology Language Overview”, s. f.).

StarUML¹⁸

Herramienta gratuita para el modelo de los componentes de software mediante UML. Esta herramienta permite el modelamiento de distintos diagramas (“StarUML 2 Documentation”, s. f.). Para el presente proyecto se diseñarán los diagramas de clases, de componentes, de casos de uso y de despliegue.

2.4.2 Metodologías

Rational Unified Process RUP

RUP es una metodología para el proceso de desarrollo de software. Provee un estándar en fases para el análisis, diseño, implementación y documentación. El presente proyecto se basará en RUP para la documentación del documento de Análisis y de Arquitectura.

¹⁸ <http://staruml.io/>

Unified Modeling Language UML

UML¹⁹ es un lenguaje de modelado de aplicaciones de software. Permite especificar y documentar a nivel gráfico cada uno de los componentes y funcionalidades de un sistema.

2.5 Alcance y Limitaciones

El proyecto forma parte del área de ciencias de la computación, y específicamente en la subárea de *linked data*. Para este proyecto se ha establecido el diseño y la implementación de un modelo que contempla la ontología y las funciones de recuperación o extracción de los datos. El contexto de la información serán las publicaciones y autores de nacionalidad peruana que estén publicadas bajo datos enlazados y que sean de libre acceso. Asimismo, para la interfaz se contará con un prototipo web, con la posibilidad de ser visualizada tanto en Chrome como Mozilla Firefox.

2.5.1 Riesgos

En la *Tabla 2.1* se presentan los riesgos identificados para el presente proyecto de tesis, así como el impacto de su ocurrencia y las medidas para mitigarlo.

Riesgo	Impacto	Medidas
Mayor complejidad en las herramientas a utilizar para el desarrollo de los entregables puede generar retrasos en el cronograma establecido para los entregables o que sean de baja calidad.	Medio	Contemplar en el cronograma el esfuerzo en horas para la curva de aprendizaje para así establecer el tiempo necesario. Buscar personas que tengan conocimientos sobre el tema y herramientas a utilizar para que puedan ayudar a absolver dudas.
Incapacidad para encontrar autores que brinden su información sobre sus publicaciones.	Alto	Elaborar un plan de acción estipulando una fecha límite adecuada (dos primeras semanas) para la búsqueda y consecución de los autores.
Actualización de los frameworks y demás herramientas que conlleven a una actualización o modificación de lo ya implementado en el proyecto (migración de versión).	Bajo	Utilizar herramientas que tengan el grado de madurez que permitan la facilidad de migración entre versiones. Estar constantemente actualizado sobre las

¹⁹ <http://www.uml.org/>

		nuevas versiones disponibles, soluciones de bugs y demás componentes sobre las herramientas a utilizar.
Problema de salud del alumno que impida cumplir con las horas establecidos de acuerdo al cronograma, generando retrasos en el proyecto	Medio	Estimar un colchón de tiempo para imprevistos e incidentes que permita avanzar el proyecto y concluir los entregables con un tiempo extra al <i>deadline</i> establecido
Viaje del asesor que impida la revisión de acuerdo a los plazos establecidos	Medio	Coordinar de forma anticipada cualquier viaje o ausencia para que se realicen las acciones correspondientes y, de ser necesario, adelantar el plazo del entregable preferentemente

Tabla 2.1 Riesgos del proyecto

2.6 Justificación de la solución

La solución propuesta busca implementar un software que permita la recuperación de la información de publicaciones académicas y científicas con un énfasis en los autores de origen peruano. Esta orientación tiene utilidad para fines académicos y de investigación tanto para los propios autores como para otros académicos, investigadores y estudiantes que necesiten información relevante sobre investigaciones peruanas ya realizadas sobre distintas áreas que sirvan de referencia para establecer nuevas investigaciones y trabajos. De esta manera, se logrará brindar visibilidad sobre las investigaciones peruanas, datos estadísticos y la información relevante sobre ellas.

Asimismo, el modelo no se restringe a un funcionamiento único para hacer búsquedas en el dominio escogido ya que, al hacer cambios en la ontología, ampliarla o quitar ciertas restricciones, se puede emplear para hacer búsquedas en otros dominios (autores de cualquier nacionalidad). De esta manera se amplía la posibilidad de encontrar mayores fuentes de información.

Finalmente, el software a implementar servirá para trabajos futuros como el de ampliar la plataforma para estructurar un repositorio digital propio con sus propios procedimientos de publicación de investigaciones y artículos, brindando una solución integral de un repositorio digital con la capacidad de realizar búsquedas sobre otros repositorios ya existentes. Así también se puede abordar la estructuración de ontologías a nivel de usuario para realizar distintos tipos de búsquedas que permitan atender distintas necesidades sobre áreas y especialidades específicas del conocimiento.

2.7 Análisis de Viabilidad

El análisis de la viabilidad estará comprendido por los enfoques técnicos, temporales, económicos y de necesidades. Mediante los cuatro factores se podrá establecer la factibilidad para la implementación del proyecto.

2.7.1 Viabilidad técnica

La experticia técnica no significará un impedimento ya que se tiene el conocimiento sobre el lenguaje de programación Java, así como del *framework* de Spring. La curva de aprendizaje estará enfocada en la tecnología que se aplica en *linked data*: RDF, SPARQL, y JENA como el *framework* para el manejo de las ontologías. JENA es un *framework* para Java, por lo que no existirán problemas de compatibilidad. Asimismo, todas las herramientas mencionadas son gratuitas, por lo que el acceso no estará limitado y existe extensa documentación sobre el uso de cada una de ellas que permitirán afrontar la curva de aprendizaje en el tiempo necesario previo a la implementación del software. De acuerdo a lo estipulado, el presente proyecto es viable a nivel técnico.

2.7.2 Viabilidad temporal

Para la implementación o consecución de cada uno de los resultados esperados se tiene un tiempo requerido que se adapta a los tiempos establecidos por el curso. Para tales motivos se presenta la estimación del tiempo para la implementación de cada resultado esperado, el cual contempla su consecución antes de la finalización del curso.

Resultado esperado	# de semanas
RE1.1 y RE1.2	3
RE2.1	3
RE3.1	3
RE4.1	2
RE4.2	2
Total	13

Tabla 2.2 Estimación del tiempo requerido

Así mismo, se presenta la estimación de horas por semana que se dedicará para el desarrollo del proyecto. De acuerdo con la *Figura 2.2* se tendrá un tiempo estimado de 16 horas por semana, resultando en un tiempo total estimado de 208 horas para las 13 semanas.

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
07:00-08:00							
08:00-09:00							
09:00-10:00							
10:00-11:00							Tesis
11:00-12:00							Tesis
12:00-13:00							
13:00-14:00							
14:00-15:00						Tesis	Tesis
15:00-16:00						Tesis	Tesis
16:00-17:00						Tesis	
17:00-18:00						Tesis	
18:00-19:00	Tesis	Tesis					
19:00-20:00	Tesis	Tesis					
20:00-21:00	Tesis	Tesis					
21:00-22:00	Tesis	Tesis					
22:00-23:00							

Figura 2.2 Estimación de horas por semana

2.7.3 Viabilidad económica

Para el presente proyecto las herramientas a utilizar son de código abierto *-open source-*, por lo cual no se incurrirán en gastos para la implementación de la herramienta de software. Asimismo, se cuenta con una laptop, así como las computadoras de la facultad como parte del hardware necesario. Por lo tanto, el proyecto es viable económicamente ya que no se necesitan componentes de hardware o software que signifiquen un gasto para el proyecto.

2.7.4 Análisis de necesidades

Un factor crítico para el proyecto es la necesidad de sacar fuentes de datos de datasets que son de libre acceso y comparar con la información que puedan brindar los autores sobre sus publicaciones, de tal manera que se pueda comparar que los resultados encontrados son los esperados y los especificados por los autores.

CAPÍTULO 3: Diseño de la ontología dinámica

En el presente capítulo se desarrollará el primer objetivo específico del proyecto, el cual trata del diseño de la ontología dinámica que permita la descripción de recursos en base a ontologías existentes en distintos repositorios digitales. Para satisfacer las necesidades, se plantea lo siguiente:

- i. Diseñar una ontología base que permita almacenar información relevante sobre autores y publicaciones.
- ii. Esta ontología podrá ser actualizada para agregar nuevas clases o propiedades que permitan almacenar otros datos relevantes.
- iii. Se implementará un algoritmo que permita la adaptación de la ontología a distintos escenarios (bases de datos) que poseen sus propias ontologías o estructuras para definir los datos (objetos, propiedades y valores).

Para el punto i, el diseño de la ontología base, se aplicará el método de Preguntas de Verificación, el cual es explicado en un marco de trabajo por Natasha Noy y Deborah McGuinness (Noy & McGuinness, s. f.).

3.1 Diseño metodológico de la ontología

Para el adecuado diseño de la ontología se debe establecer el dominio y alcance, es decir, el contexto de información donde se aplicará el modelo ontológico. Luego, se debe evaluar la utilización de una ontología ya existente o la creación de una propia.

3.1.1 Determinar el dominio y alcance de la ontología base

- ¿Cuál es el dominio de la ontología?

El dominio son los autores o investigadores peruanos y sus publicaciones.

- ¿Para qué se usará la ontología?

Para identificar los datos principales de los autores, así como los metadatos e información relevante de sus publicaciones.

- ¿Qué se podrá responder con la ontología?

Con respecto a los autores, se podrá conocer su lista de publicaciones realizadas por año, el campo de investigación del autor, la institución asociada. Para el caso de las publicaciones se podrá conocer el año de publicación, la lista de autores y co-autores, el resumen (abstract) de la publicación, publicaciones a las que cita o referencia y viceversa, si la publicación pertenece a una conferencia.

- ¿Cómo o quién usará la ontología?
La ontología formará parte de una solución técnica (software) para búsqueda de publicaciones en distintos repositorios (bases de datos externas) con tecnología de linked data.
- ¿La ontología tendrá algún mantenimiento?
La ontología podrá ser actualizada con nuevas propiedades, si fuera necesario, para obtener más información relevante.

Las preguntas de competencia permitirán conocer si la base de conocimiento que soporta la ontología podrá responder las preguntas o necesidades planteadas. Estas preguntas, que sirven para validación en pruebas posteriores, son las que se presentan en la *Tabla 3.1*:

Preguntas de verificación
¿Cómo se llama el autor?
¿Con qué otros nombres o alias está registrado el autor?
¿Cuál es el título de la publicación?
¿Qué autores tienen más publicaciones por año?
¿Qué publicaciones han sido citadas más veces?
¿Cuál es el área de investigación del autor o de la publicación?
¿Dónde puedo encontrar información del autor?

Tabla 3.1 – Preguntas de verificación

3.1.2 Considerar la reutilización de ontologías existentes

En esta sección se deben considerar ontologías existentes que podrían resolver la necesidad. Se ha identificado la siguiente relación de ontologías de dominio:

- FOAF²⁰ (Friend of a Friend): Permite la descripción de personas y relacionarlas con información presente en la web (“FOAF Vocabulary Specification”, s. f.). Esta información es un contenido social, tal como se aprecia en la *Figura 3.1*:

²⁰ <http://www.foaf-project.org/>

FOAF Core	Social Web
<ul style="list-style-type: none"> • <ul style="list-style-type: none"> ◦ Agent ◦ Person ◦ name ◦ title ◦ img ◦ depiction (depicts) ◦ familyName ◦ givenName ◦ knows ◦ based_near ◦ age ◦ made (maker) ◦ primaryTopic (primaryTopicOf) • <ul style="list-style-type: none"> ◦ Project ◦ Organization ◦ Group ◦ member • <ul style="list-style-type: none"> ◦ Document ◦ Image 	<ul style="list-style-type: none"> • nick • mbox • homepage • weblog • openid • jabberID • mbox_sha1sum • interest • topic_interest • topic (page) • workplaceHomepage • workInfoHomepage • schoolHomepage • publications • currentProject • pastProject • account • OnlineAccount • accountName • accountServiceHomepage • PersonalProfileDocument • tipjar • sha1 • thumbnail • logo

Figura 3.1: Clases y Propiedades de FOAF Ontology (“FOAF Vocabulary Specification”, s. f.)

- DC²¹ (Dublín Core): Permite modelar los términos de los metadatos de las publicaciones (“DCMI Metadata Terms”, s. f.).
- AKT²²: Ontología diseñada para describir una comunidad académica, en cuanto a las instituciones, autores y publicaciones, pero para el dominio de Ciencias de la Computación, por lo que sus clases están orientadas específicamente a dicho dominio (“Linked Open Vocabularies (LOV)”, s. f.). Clases que utiliza:
 - Docs: modela las publicaciones.
 - Events: modela los eventos.
 - Organizations: modela las personas y las organizaciones.
 - Projects: modela los proyectos.
 - Research-area: modela el área de investigación.
 - Techs: modela las tecnologías.

De acuerdo con el análisis realizado, se verifica que no existe una ontología que se adapte a las necesidades específicas del dominio. Sin embargo, existen clases y propiedades de las distintas ontologías que se pueden reutilizar. Por lo tanto, se establecerá una ontología base para el dominio, pero reutilizando clases y propiedades de otras ontologías existentes.

²¹ <http://dublincore.org/>

²² <http://aktors.org/ontology.htm>

3.1.3 Enumerar términos importantes de la ontología

Los términos importantes son los conceptos que abarcará la ontología. Cada uno de los términos, a su vez, permitirán contestar las preguntas de verificación previamente planteada. En tal sentido, el diseño de la ontología deberá abarcar la siguiente terminología escogida:

Preguntas de verificación	Términos importantes
¿Cómo se llama el autor?	Autor, Nombre
¿Con qué otros nombres o alias está registrado el autor?	Autor, Alias
¿Cuál es el título de la publicación?	Publicación, título
¿Qué autores tienen más publicaciones por año?	Autor, Publicación
¿Qué publicaciones han sido citadas más veces?	Publicación, referencias
¿Cuál es el área de investigación del autor o de la publicación?	Autor, Publicación, Área, Especialidad
¿Dónde puedo encontrar información del autor?	Autor, Homepage

Tabla 3.2 – Términos importantes

3.1.4 Definir las clases y jerarquías de clases

En esta sección se definen las clases mínimas que debe contemplar la ontología:

- Autor: Persona que publica un artículo
- Publicación: Documento publicado (artículo) por uno o más autores
- Área: rama de la ciencia a la que se dedica el autor o la que abarca el tema de la publicación
- Institución: Centro o casa de estudios del que el autor proviene y realiza actividades de investigación o al que está asociada la publicación realizada.

3.1.5 Definir las propiedades de las clases

Para cada clase se define una serie de propiedades, las cuales pueden ser de tipo Objeto o Data (literal):

Clase	Tipo Objeto	Tipo Data
Autor	centro_investigacion	nombre
	ciudad	alias
	area	fecha_nacimiento
	-	homepage

Publicación	su_autor	título
	citada_por	fecha_publicacion
	revista	resumen
	tipo	url
Area	-	nombre
Institución	-	nombre

Tabla 3.3 – Propiedades de clases

3.1.6 Definir las restricciones de las propiedades

En la presente sección se definen el dominio, rango, cardinalidad y el tipo de valor de cada una de las propiedades definidas.

Propiedad	Tipo Valor	Cardinalidad	Dominio	Rango
centro_investigacion	Instancia	Simple	Autor	Institución
ciudad	Instancia	Simple	Autor	Ciudad
area	Instancia	Simple	Autor	Area
nombre	Cadena (literal)	Simple	Cualquiera	Tipo Cadena
alias	Cadena (literal)	Simple	Autor	Tipo Cadena
fecha_nacimiento	Cadena (literal)	Simple	Autor	Tipo Cadena
homepage	Cadena (literal)	Simple	Autor	Tipo Cadena
su_autor	Instancia	Simple	Publicación	Autor
citada_por	Instancia	Simple	Publicación	Publicación
revista	Instancia	Simple	Publicación	Cualquiera
tipo	Instancia	Simple	Publicación	Cualquiera
título	Cadena (literal)	Simple	Publicación	Tipo Cadena
fecha_publicacion	Cadena (literal)	Simple	Publicación	Tipo Cadena
resumen	Cadena (literal)	Simple	Publicación	Tipo Cadena
url	Cadena (literal)	Simple	Publicación	Tipo Cadena

Tabla 3.4 – Restricciones de propiedades

3.2 Resultado de la Metodología

Mediante el análisis realizado, se obtiene la siguiente ontología base (BOAP Base Ontology Authors and Publications):

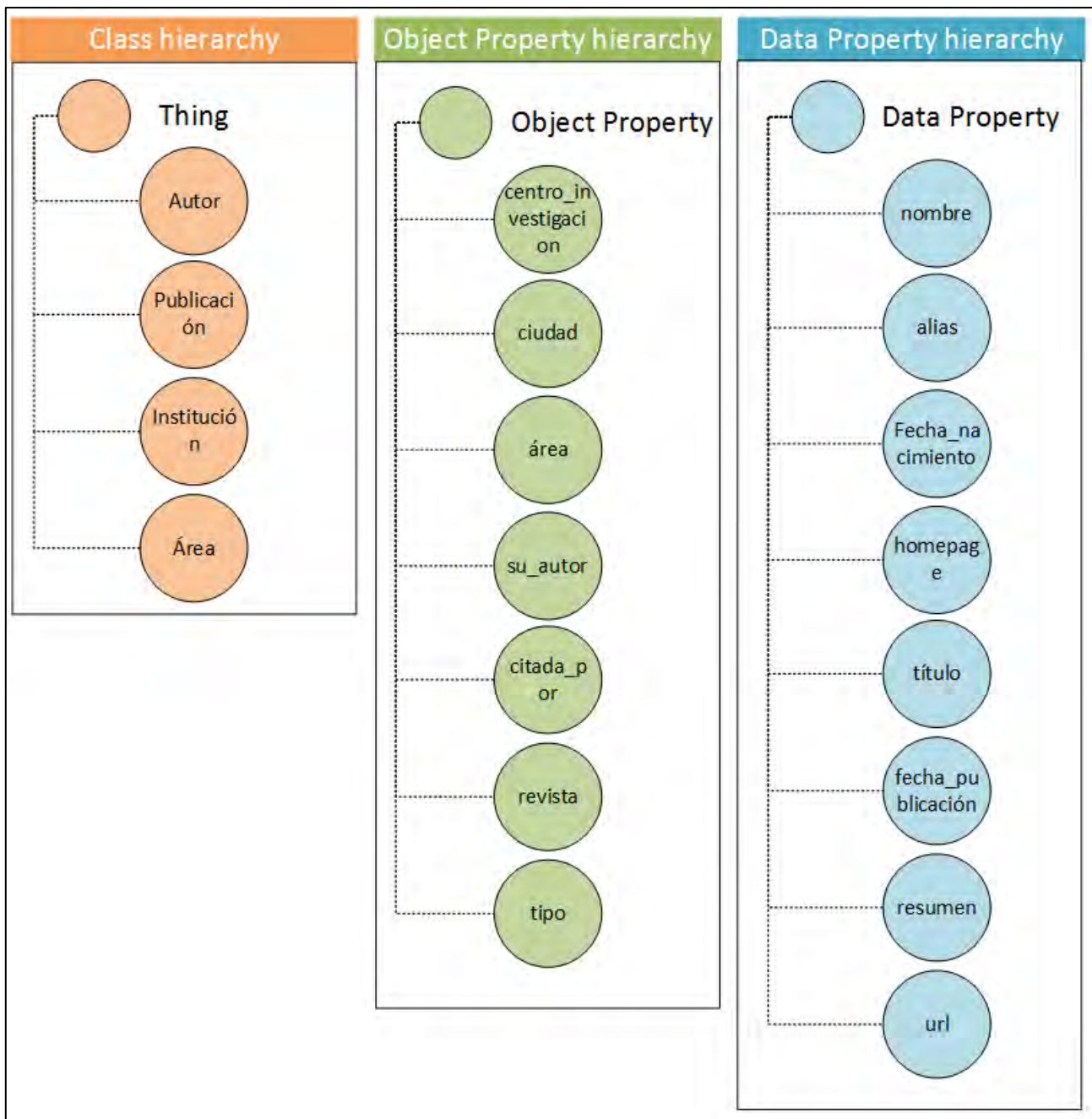


Figura 3.2 – Ontología Base (BOAP)

3.3 Validación de la ontología

Para comprobar que la ontología cumple con las necesidades básicas, se debe verificar que la ontología planteada permita responder a las preguntas de verificación previamente establecidas.

Preguntas de verificación	Términos importantes
¿Cómo se llama el autor?	Autor → Nombre → “Cadena Nombre”
¿Con qué otros nombres o alias está registrado el autor?	Autor → Alias → “Cadena Alias”
¿Cuál es el título de la publicación?	Publicación → título → “Cadena título”
¿Qué autores tienen más publicaciones por año?	Publicación → su_autor → Autor Publicación → fecha → “cadena formato fecha”
¿Qué publicaciones han sido citadas más veces?	Publicación → citada_por → Instancia Publicación
¿Cuál es el área de investigación del autor o de la publicación?	Autor → Área → Instancia Área Autor → Publicación → Instancia Área
¿Dónde puedo encontrar información del autor?	Autor → Homepage → “Cadena url”

Tabla 3.5 – Validación de Ontología

3.4 Algoritmo adaptador de ontología

En esta sección se detalla el algoritmo que permite la adaptación dinámica del algoritmo base. Dado que cada dataset cuenta con su propia ontología se debe conocer dicha especificación, así como los recursos (clases y propiedades) que utilice de otras ontologías (ontologías top-level, o de dominio).

3.4.1 Pseudocódigo principal

El *Pseudocódigo 3.1* presenta el pseudocódigo principal del algoritmo Adaptador de Ontología, que cuenta con las siguientes funciones relevantes: Leer Ontologías, Extraer Recursos, Generar Ontología y Serializar. Este algoritmo, tiene como dato de entrada a DatasetRdf, el cual es la URL del Sparql End Point o, en su defecto, la ruta (URL o local) del archivo RDF del Dataset. Este pseudocódigo es la solución para extender la ontología base en tiempo de ejecución según otras ontologías existentes mediante la extracción de aquellos recursos relevantes para el dominio.

3.1 Pseudocódigo principal del algoritmo Adaptador (DatasetRdf)

1. Inicio
 - 1.1. *Lista_Ontología = Leer_Ontologías(DatasetRdf)*
 - 1.2. *Para i = 1 Hasta (Lista_Ontología.cantidad())*
 - 1.2.1. *Lista_R = Extraer_Recursos(Lista_Ontología[i])*
 - 1.3. *Fin Para*
 - 1.4. *Onto_Dinámica = Generar_Ontología(Lista_Ontología, lista_R)*
 - 1.5. *Serializar(ArchivoRDF, Onto_Dinámica)*
2. Fin

1.1. DatasetRdf es la instancia de la base de datos de donde se extraerá la información. De este dataset, se recuperan las ontologías que utiliza. Por ejemplo, al leer el dataset de DBLP encontraremos que utiliza las ontologías: rdf, rdfs, owl, akt y akts (prefijos de las ontologías).

1.2. Recorrido de todas las ontologías

1.2.1. Se extraen los recursos (clases y propiedades) que son relevantes para el dominio (publicaciones y autores) y se agregan a la lista.

1.3. Fin del recorrido

1.4. Se genera la nueva ontología en base a los recursos extraídos y la lista de las ontologías de origen.

1.5. Se convierte la estructura ontológica RDF en un archivo RDF que permitirá la carga de datos o información.

3.4.2 Leer Ontologías

Esta función permitirá conocer qué ontologías utiliza un dataset RDF; es decir, se podrá conocer qué esquemas de vocabularios utiliza para la definición de sus clases y propiedades. Estos esquemas, son los que usa el dataset para estructurar su información en grafos RDF. El dataset también podría tener su propio esquema. Por ejemplo, DBLP, aparte de lo mencionado previamente, también hace uso de su esquema cuyo prefijo es dblp, esto le permite extender su base ontológica con otras clases y propiedades propias de su dominio. La función devolverá una lista de prefijos con su URL: [owl http://www.w3.org/2002/07/owl#].

3.2 Pseudocódigo Leer_Ontologías

1. Inicio Leer_Ontologías(DatasetRDF)
 - 1.1. Si EsArchivo(DatasetRDF) Hacer
 - 1.1.1. AbrirArchivo(DatasetRDF)
 - 1.1.2. línea=Leer (DatasetRDF)
 - 1.1.3. Mientras esPrefijo(línea) Hacer
 - 1.1.3.1. Lista_Prefijo = agregarPrefijo(línea)
 - 1.1.3.2. línea=Leer(DatasetRDF)
 - 1.1.4. Fin Mientras
 - 1.2. Sino Si Endpoint(DatasetRDF)
 - 1.2.1. Instancia = Concetar_Endoint(DatasetRDF)
 - 1.2.2. Lista_Prefijo = Obtener_Prefijos(Instancia)
 - 1.3. Fin Sino
 - 1.4. Retornar Lista_Prefijo
2. Fin

- 1.1. Si se trata de un archivo en formato RDF/XML
 - 1.1.1. Se abre el archivo RDF/XML
 - 1.1.2. Se lee una línea del archivo
 - 1.1.3. Mientras la línea contenga la definición de un prefijo. Por ejemplo:
xmlns:owl="http://www.w3.org/2002/07/owl#"
 - 1.1.3.1. Se agrega el prefijo a la lista
 - 1.1.3.2. Se lee siguiente línea
 - 1.1.4. Fin Mientras
 - 1.2. Si se trata de un SPARQL End Point (URL del servicio)
 - 1.2.1. Se obtiene la conexión al servicio
 - 1.2.2. Se obtienen los prefijos mediante consulta SPARQL
 - 1.3. Fin
 - 1.4. Se retorna la lista de prefijos

3.4.3 Extraer Recursos

Mediante la lista de prefijos, se accede a las ontologías de cada una (ubicadas en Internet o ya descargadas localmente). Debido a esto, se tiene una restricción: **es necesario poder obtener la**

definición de la ontología, caso contrario no se pueden extraer los recursos y se devuelve una lista vacía.

3.3 Pseudocódigo Extraer Recursos

1. *Inicio Extraer_Recursos(prefijoURI)*
 - 1.1. *Archivo Ontología = Buscar Ontología(prefijoURI)*
 - 1.2. *Lista Recursos = Ejecutar SPARQL(Archivo Ontología)*
 - 1.3. *Para i=1 Hasta (Nro. Recursos) Hacer*
 - 1.3.1. *Si Es Relevante(recurso)*
 - 1.3.1.1. *Lista_R = Agregar(recurso)*
 - 1.3.2. *Fin Si*
 - 1.4. *Fin Para*
 - 1.5. *Retornar Lista_R*
2. *Fin*

- 1.1. Se accede a la definición de la Ontología en RDF/XML
- 1.2. Se ejecuta consulta para extraer los recursos (clases y propiedades)
- 1.3. Para cada recurso encontrado
 - 1.3.1. Se verifica si es relevante, es decir, si el recurso forma parte del dominio de autores y publicaciones: la propiedad “dc:title” sería relevante, mientras la propiedad “foaf:account” no lo sería.*
 - 1.3.1.1 Se agrega el recurso a la lista relevante.
 - 1.4. Fin Para
 - 1.5. Se retorna la lista de recursos relevantes

*Para reconocer si un recurso (clase o propiedad) es relevante se hará uso de una lista de “palabras claves” para autores y publicaciones que formarán un corpus actualizable. Esta lista de palabras claves, a su vez, estará asociado a una serie de propiedades previamente catalogadas a partir de ontologías bases como rdf, rdfs, owl, foaf, y akt. El detalle de dicho análisis será presentado en el acápite **3.3.6 Generación y mantenimiento del corpus de palabras claves.**

3.4.4 Generar Ontología

Esta función permitirá la creación de un grafo RDF que contenga la definición de la Ontología para el dominio de autores y publicaciones. El grafo por generar será una extensión del Grafo Base del dominio.

3.4 Pseudocódigo Generar Ontología

1. Inicio *Generar_Ontología(Lista_Prefijo, Lista_R)*
 - 1.1. *Grafo Ontología = Inicializar(Ontología Base)*
 - 1.2. *Para i = 1 Hasta (Nro. Prefijos)*
 - 1.2.1. *Nodo = Crear Nodo(prefijo[i])*
 - 1.2.2. *Grafo Ontología = agregar Prefijo(Nodo)*
 - 1.3. *Fin para*
 - 1.4. *Para i = 1 Hasta (Nro. Recursos)*
 - 1.4.1. *Nodo = Crear Nodo(recurso[i])*
 - 1.4.2. *Grafo Ontología = agregar Recurso(Nodo)*
 - 1.5. *Fin Para*
 - 1.6. *Retornar Grafo Ontología*
2. *Fin*

- 1.1. Se inicializa un grafo RDF con la Ontología Base
- 1.2. Recorrido de los prefijos de las ontologías
 - 1.2.1. Se crea un Nodo con el prefijo
 - 1.2.2. Se agrega el Nodo al Grafo Ontología
- 1.3. Fin del recorrido
- 1.4. Recorrido de los recursos
 - 1.4.1. Se crea un Nodo con el recurso
 - 1.4.2. Se agrega el Nodo al Grafo Ontología
- 1.5. Fin del recorrido
- 1.6. Se retorna el grafo generado

3.4.5 Serializar

Esta función permitirá convertir el grafo RDF en un archivo RDF/XML para su posterior utilización.

3.5 Pseudocódigo Serializar

1. Inicio Serializar(*Archivo RDF, Ontología*)
 - 1.1. Abrir Archivo(*Archivo RDF*)
 - 1.2. Escribir(*Ontología*)
 - 1.3. Cerrar Archivo(*Archivo RDF*)
2. Fin

- 1.1. Apertura del archivo
 - 1.2. Escritura en el archivo
 - 1.3. Clausura del archivo.
-

3.4.6 Generación y mantenimiento del corpus de palabras claves

En la presente sección se detallará la lógica para el uso de palabras claves y el preprocesamiento realizado a las ontologías top-level y de dominio que, en muchas situaciones, los datasets toman como base para el vocabulario ontológico:

- a. Se generó una lista de palabras clave para Autores y Publicaciones de acuerdo con la *Figura 3.3*. Esta lista de palabras claves permitirá definir y delimitar el contexto de la ontología para la búsqueda de metadatos. Asimismo, las palabras claves estarán divididas por un grupo de términos relacionados al Autor y otro grupo de términos relacionados a la Publicación.
- b. Se consideraron como ontologías top-level y de dominio la relación presentada en la *Figura 3.4*.
- c. De acuerdo con dicha relación, se realizó un análisis para reconocer aquellas propiedades que son relevantes para el dominio. Por ejemplo:
 - En la *Figura 3.5* se observa que sólo la propiedad “rdf:type” es relevante.
 - Para la *Figura 3.6* las propiedades rdfs relevantes son: Literal, label, comment y seeAlso.
 - Para la *Figura 3.7* las propiedades OWL²³ (Ontology Web Language) relevantes son: sameAs, Person, hasAge, PersonAge, versionInfo, PriorVersion.
 - Para la *Figura 3.8* las propiedades Dublin Core: abstract, bibliographicCitation, created, creator, date, description, hasVersion, isVersionOf, publisher, references, title, type.

²³ <https://www.w3.org/OWL/>

- d. Con el análisis realizado se produjo un preprocesamiento para obtener un vocabulario ontológico relacionado al corpus de keywords. Para otras ontologías, el algoritmo realizará el mismo análisis para encontrar nuevas propiedades relevante.
- e. El preprocesamiento explicado será almacenado en una base de datos relacional para mantener y actualizar dicha información. Asimismo, dicha base de datos contará con información base para las funcionalidades a implementar.

Tipo	Keyword	Tipo	Keyword
Autor	author	Publicación	article
	name		title
	FN		has-title
	fullname		name
	full-name		label
	label		booktitle
	same		author
	sameAs		has-author
	same-as		creator
	has-article		maker
	date		edited
	has-date		edited-by
	date		reference
	year		cited
	web		cited-by
	homepage		citedBy
has-web	pages		
has-web-address	has-date		
	date		
	year		
	abstract		
	has-abstract		
	resume		
	has-resume		
	homepage		
	has-web		
	has-web-address		

Figura 3.3 – Keywords

Nombre	Prefijo	Espacio de nombres
Resource Description Framework	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
Resource Description Framework Schema	rdfs	http://www.w3.org/2000/01/rdf-schema#
Ontology Web Language	owl	http://www.w3.org/2002/07/owl#
Friend of a Friend	foaf	http://xmlns.com/foaf/0.1/
XML Schema Datatype	xsd	XML Schema Datatype
Dublin Core	dc	http://purl.org/dc/elements/1.1/
Dublin Core Terms	dcterms, dct	http://purl.org/dc/terms/
Semantic Web for Research Communities	swrc	http://www.w3.org/1999/02/22-rdf-syntax-ns#
AKT Reference Ontology	akt	http://www.aktors.org/ontology/portal#
AKT Reference Support Ontology	akts	http://www.aktors.org/ontology/support#
Semantic Web Conference	swc	http://data.semanticweb.org/ns/swc/swc_2009-05-09.html#
European Semantic Web Conference	eswc	http://www.eswc2006.org/technologies/ontology#
Simple Knowledge Organization System	skos	https://www.w3.org/2009/08/skos-reference/skos.html#

Figura 3.4 – Espacios de nombres

rdf		
type	vocabulary	comment
Class	rdf:langString	The class of language-tagged string literal values.
Class	rdf:HTML	The class of HTML literal values.
Class	rdf:XMLLiteral	The class of XML literal values.
Class	rdf:Property	The class of RDF properties.
Property	rdf:type	The subject is an instance of a class.
Container Class	rdf:Bag	The class of unordered containers.
Container Class	rdf:Seq	The class of ordered containers.
Container Class	rdf:Alt	The class of containers of alternatives.
Collection Class	rdf:List	The class of RDF Lists.
Collection Property	rdf:first	The first item in the subject RDF list.
Collection Property	rdf:rest	The rest of the subject RDF list after the first item.
Collection Property	rdf:nil	
Reification Vocabulary	rdf:Statement	The class of RDF statements.
Reification Vocabulary	rdf:subject	The subject of the subject RDF statement.
Reification Vocabulary	rdf:predicate	The predicate of the subject RDF statement.
Reification Vocabulary	rdf:object	The object of the subject RDF statement.
Utility Property	rdf:value	Idiomatic property used for structured values.

Figura 3.5 – Vocabulario rdf ("RDF Schema 1.1", s. f.)

rdfs		
type	vocabulary	comment
Class	rdfs:Resource	The class resource, everything.
Class	rdfs:Class	The class of classes.
Class	rdfs:Literal	The class of literal values, e.g. textual strings and integers.
Class	rdfs:Datatype	The class of RDF datatypes.
Property	rdfs:range	A range of the subject property.
Property	rdfs:domain	A domain of the subject property.
Property	rdfs:subClassOf	The subject is a subclass of a class.
Property	rdfs:subPropertyOf	The subject is a subproperty of a property.
Property	rdfs:label	A human-readable name for the subject.
Property	rdfs:comment	A description of the subject resource.
ContainerClass	rdfs:Container	The class of RDF containers.
ContainerClass	rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
ContainerClass	rdfs:member	A member of the subject resource.
Utility Property	rdfs:seeAlso	Further information about the subject resource.
Utility Property	rdfs:isDefinedBy	The definition of the subject resource.

Figura 3.6 – Vocabulario rdfs (“RDF Schema 1.1”, s. f.)

owl	
type	vocabulary
Class Hierarchy	owl:Class
Class Hierarchy	owl:equivalentClass
Class Disjointness	owl:AllDisjointClasses
Class Disjointness	owl:members
Object Property	owl:NegativePropertyAssertion
Object Property	owl:sourceIndividual
Object Property	owl:assertionProperty
Object Property	owl:targetIndividual
Object Property	owl:ObjectProperty
Equality or Inequality	owl:differentFrom
Equality or Inequality	owl:sameAs
Datatype	owl:targetValue
Datatype	owl:DataTypeProperty
Complex Class	owl:intersectionOf
Complex Class	owl:unionOf
Complex Class	owl:complementOf
Property Restrictions	owl:Restriction
Property Restrictions	owl:onProperty
Property Restrictions	owl:someValuesFrom
Property Restrictions	owl:allValuesFrom
Property Restrictions	owl:hasValue
Property Restrictions	owl:hasSelf
Property Cardinality Restrictions	owl:maxQualifiedCardinality
Property Cardinality Restrictions	owl:minQualifiedCardinality
Property Cardinality Restrictions	owl:qualifiedCardinality
Property Cardinality Restrictions	owl:cardinality
Enumeration of Individuals	owl:oneOf
Property Characteristic	owl:inverseOf
Property Characteristic	owl:SymmetricProperty
Property Characteristic	owl:AsymmetricProperty
Property Characteristic	owl:propertyDisjointWith
Property Characteristic	owl:ReflexiveProperty
Property Characteristic	owl:IrreflexiveProperty
Property Characteristic	owl:FunctionalProperty
Property Characteristic	owl:InverseFunctionalProperty
Property Characteristic	owl:TransitiveProperty
Property Chains	owl:propertyChainAxiom
Key	owl:hasKey
Key	owl:DataProperty
Advanced Use	owl:onDatatype
Advanced Use	owl:withRestrictions
Advanced Use	owl:datatypeComplementOf
Entity	owl:Axiom
Entity	owl:annotatedSource

owl	
type	vocabulary
Entity	owl:annotatedProperty
Entity	owl:annotatedTarget
Entity	owl:imports
Entity	owl:Thing
Entity	owl:MaxCardinality
Entity	owl:MinCardinality
Class	owl:Person
Class	owl:Woman
Class	owl:Parent
Class	owl:Father
Class	owl:Mother
Class	owl:SocialRole
Class	owl:Man
Class	owl:Teenager
Class	owl:ChildlessPerson
Class	owl:Human
Class	owl:Female
Class	owl:HappyPerson
Class	owl:JohnsChildren
Class	owl:NarcisticPerson
Class	owl:MyBirthdayGuests
Class	owl:Dead
Class	owl:Orphan
Class	owl:Adult
Class	owl:YoungChild
Object Property	owl:hasWife
Object Property	owl:hasChild
Object Property	owl:hasDaughter
Object Property	owl:loves
Object Property	owl:hasSpouse
Object Property	owl:hasGrandparent
Object Property	owl:hasParent
Object Property	owl:hasBrother
Object Property	owl:hasUncle
Object Property	owl:hasSon
Object Property	owl:hasAncestor
Object Property	owl:hasHusband
Data Property	owl:hasAge
Data Property	owl:hasSSN
Datatype	owl:personAge
Datatype	owl:minorAge
Datatype	owl:majorAge
Datatype	owl:toddlerAge
Object Property	owl:Nothing
Object Property	owl:versionInfo
Object Property	owl:deprecated
Object Property	owl:backwardCompatibleWith
Object Property	owl:incompatibleWith
Object Property	owl:priorVersion

Figura 3.7 – Vocabulario owl (W3C, 2012)

dcterms		dcterms	
type	vocabulary	type	vocabulary
Property	dct:abstract	Property	dct:isPartOf
Property	dct:accessRights	Property	dct:isReferencedBy
Property	dct:accrualMethod	Property	dct:isReplacedBy
Property	dct:accrualPeriodicity	Property	dct:isRequiredBy
Property	dct:accrualPolicy	Property	dct:issued
Property	dct:alternative	Property	dct:isVersionOf
Property	dct:audience	Property	dct:language
Property	dct:bibliographicCitation	Property	dct:license
Property	dct:conformsTo	Property	dct:mediator
Property	dct:contributor	Property	dct:medium
Property	dct:coverage	Property	dct:modified
Property	dct:created	Property	dct:provenance
Property	dct:creator	Property	dct:publisher
Property	dct:date	Property	dct:references
Property	dct:dateAccepted	Property	dct:relation
Property	dct:dateCopyrighted	Property	dct:replaces
Property	dct:dateSubmitted	Property	dct:requires
Property	dct:description	Property	dct:rights
Property	dct:educationLevel	Property	dct:rightsHolder
Property	dct:extent	Property	dct:source
Property	dct:format	Property	dct:spatial
Property	dct:hasFormat	Property	dct:subject
Property	dct:hasPart	Property	dct:tableOfContents
Property	dct:hasVersion	Property	dct:temporal
Property	dct:identifier	Property	dct:title
Property	dct:instructionalMethod	Property	dct:type
Property	dct:isFormatOf	Property	dct:valid

Figura 3.8 – Vocabulario Dublín Core (“DCMI Metadata Terms”, s. f.)

CAPÍTULO 4: Diseño de métodos de extracción

En el presente capítulo se desarrollará el segundo objetivo específico del proyecto, el cual consiste en la implementación de un método de extracción de datos de *Linked Data*. Asimismo, el método de extracción consiste en una serie de funcionalidades que forma parte de un módulo ETL (extracción, transformación y carga); es decir, se podrá realizar las 3 fases según lo siguiente:

- **Extracción:** funciones en SPARQL que consulten información a los distintos datasets que contengan información sobre los autores y sus publicaciones.
- **Transformación:** los datos obtenidos serán procesados y reestructurados de acuerdo a la ontología extendida de BOAP (extendida, en cuanto que la ontología puede ser ampliada o modificada mediante las funciones de adaptación presentadas en el capítulo anterior).
- **Carga:** La información reestructurada será serializada archivos con formato RDF/XML y Turtle. Dicha información, a su vez, permitirá generar cálculos estadísticos básicos sobre los autores.

El presente capítulo sólo se centra en la fase de **Extracción**.

4.1 Pseudocódigo Principal

A continuación, se define el pseudocódigo principal, el cual consta de la creación de queries (consultas) sobre los autores y sus publicaciones en base a los recursos disponibles en el Dataset. Esta funcionalidad hace uso, previamente, del algoritmo adaptador de ontología explicado en el *Pseudocódigo 3.1* del capítulo anterior. La búsqueda de la información de autores y sus publicaciones se hace de acuerdo con el nombre o posibles alias que pueda tener el autor.

4.1 Pseudocódigo Principal

```
1. Inicio
  1.1. ArchivoRDF, Onto_Dinámica = Adaptador(URI Dataset)
  1.2. Lista_P = ListarPropiedades(Onto_Dinámica)
  1.3. Para i = 1 Hasta (Nro. Autores)
    1.3.1. Lista_UA = Obtener URI Autor(URI Dataset, Lista_P,
      autor[i])
    1.3.2. Lista_TA = Obtener Tuplas(URI Dataset, Lista_UA)
    1.3.3. Para j=1 Hasta (Lista_U.cantidad())
      1.3.3.1. Lista_UP = Obtener URI Publicaciones(URI
        Dataset, Lista_UA, Lista_P)
```

1.3.3.2. *Lista_TP = ObtenerTuplas(URI Dataset,
lista_UP)*

1.3.4. *Fin Para*

1.4. *Fin Para*

1.5. *Transformar(Onto_Dinámica, Lista_TA, Lista_TP)*

2. *Fin*

1.1. Se obtiene el archivo RDF y la Ontología generada para el dataset a consultar.

1.2. Se listan las propiedades de la ontología

1.3. Recorrido de todos los autores a consultar

1.3.1. Se extraen las URI que representan al autor

1.3.2. Se extraen todas las Tuplas (propiedades y valores) del autor

1.3.3. Recorrido para cada URI del autor

1.3.3.1. Se obtienen las URI de las publicaciones

1.3.3.2. Se obtienen las Tuplas de las publicaciones

1.3.4. Fin del recorrido

1.4. Fin del recorrido

1.5. Se llama a la función de Transformación*

*La función de transformación es especificada en el Capítulo 5 debido a que está dentro del alcance del siguiente objetivo.

4.2 Pseudocódigo Obtener URI Autor

Para recuperar las URI que referencian a un autor es necesario generar una consulta en Sparql que permita obtener las URI del autor a partir del nombre o alias que pueda tener.

4.2 Pseudocódigo Obtener URI Autor

1. *Inicio*

1.1. *Propiedades_nombre = EncontrarPropiedades(Lista_P)*

1.2. *Para i = 1 Hasta (Nro. Propiedades de nombre)*

1.2.1. *Lista_U = ObtenerURI(URI Dataset, propiedad)*

1.3. *Fin Para*

1.4. *Retornar Lista_U*

2. *Fin*

1.1. Se buscan los recursos que referencian a la propiedad Nombre, por ejemplo: foaf:name, akt:full-name.

1.2. Recorrido para cada propiedad de nombre

1.2.1. Se agregan a la lista las URI encontradas

1.3. Fin del recorrido

1.4. Se retorna la lista de URI

El *Pseudocódigo 4.3* establece la lógica para encontrar las propiedades relacionadas al recurso o URI que representa al autor.

4.3 Pseudocódigo Encontrar Propiedades

1. *Inicio*

1.1. *Para i = 1 Hasta (Nro. Propiedades)*

1.1.1. *Si propiedad[i] pertenece a recurso_nombre Hacer*

1.1.1.1. *Lista_RN = agregar propiedad[i]*

1.1.2. *Fin Si*

1.2. *Fin Para*

1.3. *Retornar Lista_RN*

2. *Fin*

1.1. Recorrido para cada propiedad

1.1.1. Si la propiedad es un recurso (predicado) cuyo valor es un nombre (objeto) se agrega a la lista de recursos de nombre.

1.2. Fin del recorrido

1.3. Se retorna la lista de recursos de nombre

El *Pseudocódigo 4.4* establece el recorrido para obtener las URI de cada una de las propiedades encontradas. Esta solución consiste en la creación en tiempo de ejecución del sparql, la conexión con el *End Point* del *Dataset* a consultar, y la respectiva ejecución de la consulta.

4.4 Pseudocódigo Obtener URI

1. Inicio
 - 1.1. *query* = Crear Query(propiedad, nombre_autor, alias_autor)
 - 1.2. Instancia = Conectar End Point (URI Dataset)
 - 1.3. Lista_U = ejecutar SPARQL(Instancia, query)
 - 1.4. Retornar Lista_U
2. Fin

- 1.1. Se crea el query en base a la propiedad*
- 1.2. Se realiza la conexión al servicio de Sparql End Point
- 1.3. Se ejecuta la consulta en el Sparql End Point y se agrega el resultado a la lista de URI
- 1.4. Se retorna la lista de URI

*La creación del query debe dar un resultado como el siguiente ejemplo para DBLP:

SPARQL Buscar URI Autor

```
PREFIX akt: <http://www.aktors.org/ontology/portal#>  
SELECT ?author  
WHERE  
{  
  ?author akt:full-name > ?fullname  
  FILTER (?fullname = "Manuel Tupia Anticona" || ?fullname = "Manuel Tupia")  
}
```

4.3 Pseudocódigo Obtener Tuplas

El Pseudocódigo 4.5 establece la lógica para la obtención de las tuplas relacionadas a las URI de un autor.

4.5 Pseudocódigo Obtener Tuplas

1. Inicio
 - 1.1. Instancia = Conectar End Point (URI Dataset)
 - 1.2. Para $i=1$ hasta Lista_U.cantidad() Hacer
 - 1.2.1. query = Crear Query(Lista_U[i])
 - 1.2.2. Lista_T = ejecutar SPARQL(Instancia, query)
 - 1.3. Fin Para
 - 1.4. Retornar Lista_T
2. Fin

- 1.1. Se realiza la conexión al servicio de Sparql End Point
- 1.2. Recorrido por cada URI
 - 1.2.1. Se crea el query para buscar las Tuplas*
 - 1.2.2. Se ejecuta la consulta en el Sparql End Point y se agrega el resultado a la lista de tuplas
- 1.3. Se retorna la lista de tuplas

*La creación del query debe dar un resultado como el siguiente ejemplo para DBLP:

SPARQL Buscar Tuplas

```
PREFIX akt: <http://www.aktors.org/ontology/portal#>

SELECT distinct ?predicate ?object

WHERE

{

  <http://dblp.uri.resource> ?predicate ?object

}
```

4.4 Pseudocódigo Obtener URI Publicaciones

Para obtener las URI que referencian a una publicación se debe conocer la URI del autor.

4.6 Pseudocódigo Obtener URI Publicación

```
1. Inicio
  1.1. Propiedades_su_autor = EncontrarPropiedades(Lista_P)
  1.2. Para i = 1 Hasta Lista_UA.cantidad() Hacer
    1.2.1. Para i = 1 Hasta (Nro. Propiedades de su_autor)
      1.2.1.1. Lista_UP = ObtenerURI(URI Dataset,
        propiedad, Lista_UA[ i ])
    1.2.2. Fin Para
  1.3. Fin Para
  1.4. Retornar Lista_UP
2. Fin
```

1.1. Se buscan los recursos que referencian a la propiedad *su_autor*, por ejemplo: *akt:has-author*, *dcterms:creator*

1.2. Recorrido para cada URI del autor

1.2.1. Recorrido para cada propiedad de *su_autor*

1.2.1.1. Se agregan a la lista las URI encontradas

1.3. Fin del recorrido

1.4. Se retorna la lista de URI

El *Pseudocódigo 4.7* define el proceso para encontrar las propiedades que referencian a la propiedad que indica la autoría del autor, es decir, aquellas propiedades que relacionen a una publicación con su respectivo(s) autor(s).

4.7 Pseudocódigo Encontrar Propiedades

```
3. Inicio
  3.1. Para i = 1 Hasta (Nro. Propiedades)
    3.1.1. Si propiedad pertenece a recurso_su_autor Hacer
      3.1.1.1. Lista_R = agregar propiedad[i]
```


- 3.1.2. *Fin Si*
- 3.2. *Fin Para*
- 3.3. *Retornar Lista_R*
- 4. *Fin*

1.4. Recorrido para cada propiedad

1.4.1. Si la propiedad es un recurso (predicado) cuyo valor es una instancia de un autor (objeto) se agrega a la lista de recursos de su_autor.

1.5. Fin del recorrido

1.6. Se retorna la lista de recursos de su_autor

El *Pseudocódigo 4.8* define el proceso para encontrar las URI de las publicaciones cuya autoría es el autor consultado. De esta manera, se logra encontrar la lista de las publicaciones relacionadas a un autor específico. Este proceso consiste en la creación de la consulta Sparql, la conexión con el *End Point* del *Dataset* a consultar, la ejecución de la consulta y el almacenamiento de la lista de resultados (tuplas extraídas).

4.8 Pseudocódigo Obtener URI (URI Dataset, propiedad, URI_autor)

- 3. *Inicio*
 - 3.1. *query = Crear Query(propiedad, URI_autor)*
 - 3.2. *Instancia = Conectar End Point (URI Dataset)*
 - 3.3. *Lista_U = ejecutar SPARQL(Instancia, query)*
 - 3.4. *Retornar Lista_U*
- 4. *Fin*

1.5. Se crea el query en base a la propiedad y el URI del autor*

1.6. Se realiza la conexión al servicio de Sparql End Point

1.7. Se ejecuta la consulta en el Sparql End Point y se agrega el resultado a la lista de URI

1.8. Se retorna la lista de URI

*La creación del query debe dar un resultado como el siguiente ejemplo para DBLP:

SPARQL Buscar URI Publicación

```
PREFIX akt: http://www.aktors.org/ontology/portal#  
  
SELECT distinct ?publication  
  
WHERE  
  
{  
  
  ?publication akt:has-author <http://dblp.uri.autor>  
  
}
```



CAPÍTULO 5: Diseño de métodos de transformación

El presente capítulo tiene como finalidad dar solución al tercer objetivo específico del proyecto, el cual consiste en generar un método de transformación para los datos leídos desde uno o más datasets RDF.

5.1 Pseudocódigo Principal

El presente algoritmo tiene como parámetros de entrada a:

- Onto_Dinámica: La ontología dinámica generada previamente
- Lista_TA: Lista de tripletas de autores
- Lista_TP: Lista de tripletas de publicaciones

5.1 Pseudocódigo Principal

```
1. Inicio
  1.1. GrafoRDF = Inicializar(Onto_Dinámica)
  1.2. Para i = 1 Hasta Lista_TA.cantidad()
    1.2.1.propiedad = ObtenerPropiedad(Lista_TA[i])
    1.2.2.Si propiedad pertenece a Onto_Dinámica Entonces
      1.2.2.1. GrafoRDF = Agregar(Lista_TA[i])
    1.2.3.Sino
      1.2.3.1. Onto_Dinámica = Agregar(propiedad)
      1.2.3.2. GrafoRDF = Actualizar(Onto_Dinámica)
      1.2.3.3. GrafoRDF = Agregar(Lista_TA[i])
    1.2.4.Fin Sino
  1.3. Fin Para
  1.4. Para i = 1 Hasta Lista_TP.cantidad()
    1.4.1.propiedad = ObtenerPropiedad(Lista_TP[i])
    1.4.2.Si propiedad pertenece a Onto_Dinámica Entonces
      1.4.2.1. GrafoRDF = Agregar(Lista_TP[i])
    1.4.3.Sino
      1.4.3.1. Onto_Dinámica = Agregar(propiedad)
      1.4.3.2. GrafoRDF = Actualizar(Onto_Dinámica)
      1.4.3.3. GrafoRDF = Agregar(Lista_TP[i])
    1.4.4.Fin Sino
```

1.5. *Fin Para*

1.6. *Serializar(Archivo RDF, GrafoRDF)*

2. *Fin*

1.1. Se inicializa un grafo RDF que tendrá como vocabulario ontológico los recursos de Onto_Dinámica

1.2. Recorrido de todas las tripletas del autor

1.2.1. Se obtiene la propiedad de la tripleta

1.2.2. Si la propiedad pertenece a la Ontología Dinámica

1.2.2.1. Agregar la tripleta al grafo RDF

1.2.3. Sino si la propiedad no pertenece a la Ontología Dinámica

1.2.3.1. Agregar la propiedad a la Ontología Dinámica

1.2.3.2. Actualizar el grafoRDF con la Ontología Dinámica

1.2.3.3. Agregar la tripleta al grafoRDF

1.2.4. Fin Sino

1.3. Fin del recorrido

1.4. Recorrido de todas las tripletas de publicaciones

1.4.1. Se obtiene la propiedad de la tripleta

1.4.2. Si la propiedad pertenece a la Ontología Dinámica

1.4.2.1. Agregar la tripleta al grafo RDF

1.4.3. Sino si la propiedad no pertenece a la Ontología Dinámica

1.4.3.1. Agregar la propiedad a la Ontología Dinámica

1.4.3.2. Actualizar el grafoRDF con la Ontología Dinámica

1.4.3.3. Agregar la tripleta al grafoRDF

1.4.4. Fin Sino

1.5. Fin del recorrido

1.6. Se serializa el grafo RDF y se genera un archivo RDF/XML

CAPÍTULO 6: Diseño de la interfaz web

En el presente capítulo se desarrolla el cuarto objetivo específico del proyecto. Se trata del diseño y desarrollo de la interfaz web que permita la interacción del usuario con cada una de las funcionalidades desarrolladas previamente en los objetivos anteriores. En tal sentido los entregables claves son el documento de Análisis y documento de Arquitectura de software (adjuntados en la sección de Anexos).

6.1 Metodología para la gestión y desarrollo del proyecto

Ya que el producto consiste en un prototipo, se plantea sólo la documentación necesaria para la comprensión de la solución:

- Lista de requisitos del software
- Documento de Análisis
- Documento de Arquitectura

6.2 Requisitos del Software

Nº	Descripción	Tipo	Prioridad	Módulo	Caso de uso asociado
1	El sistema deberá permitir al administrador agregar, modificar, o anular perfiles.	F	3	Administración del Sistema	Mantener perfiles
2	El sistema deberá permitir al administrador agregar, modificar, o anular usuarios.	F	3	Administración del Sistema	Mantener usuarios
3	El sistema deberá permitir al especialista agregar, modificar, o anular instituciones.	F	2	Mantenimiento de Maestros	Mantener instituciones
4	El sistema deberá permitir al especialista agregar, modificar, o anular autores.	F	2	Mantenimiento de Maestros	Mantener autores
5	El sistema deberá permitir al especialista agregar, modificar, o anular datasets.	F	2	Mantenimiento de Maestros	Mantener datasets
6	El sistema deberá permitir al especialista agregar, modificar, o anular ontologías.	F	2	Mantenimiento de Maestros	Mantener ontologías
7	El sistema deberá permitir al especialista agregar, modificar, o anular vocabulario ontológico.	F	2	Mantenimiento de Maestros	Mantener vocabulario ontológico
8	El sistema deberá permitir al especialista agregar, modificar, o anular palabras clave.	F	2	Mantenimiento de Maestros	Mantener corpus de palabras clave
9	El sistema deberá permitir al investigador realizar consultas sobre las publicaciones de investigadores.	F	1	Consultas	Consultar publicaciones
10	El sistema deberá permitir al investigador generar reportes de estadísticas sobre las publicaciones y sus autores.	F	1	Consultas	Consultar estadísticas
11	El sistema deberá permitir al investigador realizar búsquedas generales sobre publicaciones o autores, a partir de un sparql endpoint o de archivo rdf del dataset y su respectiva ontología.	F	1	Consultas	Realizar búsquedas generales
12	El sistema debe poder ser visualizado en Google Chrome o Mozilla Firefox	NF		-	-

Figura 6.1 Catálogo de requisitos

Tipo

Valores	Descripción
F	Funcional
NF	No Funcional

Imp: Importancia/Prioridad

Valores	Descripción
1	Alta
2	Media
3	Baja

6.3 Análisis

La solución planteada requiere el soporte de las siguientes entidades: Usuario, Autor, Alias, Tipo de institución, Institución, Dataset, Ontología, Vocabulario Ontológico, Palabras Claves. Mayor detalle se puede ver en el documento de Análisis en la sección de Anexos.

6.4 Arquitectura de Software

En la *Figura 6.2* se aprecian los componentes a utilizar del Framework Spring y de Jena. El Framework de Spring permitirá el desarrollo de la interfaz web mediante el patrón de arquitectura MVC (Modelo-Vista-Controlador). Por otro lado, el Framework de Jena brinda el API para el manejo de ontologías, grafos RDF y de la utilización de consultas en SPARQL. Asimismo, la presente solución contará con una Base de Datos en MySQL. Mayor detalle se puede ver en el documento de Arquitectura en la sección de Anexos.

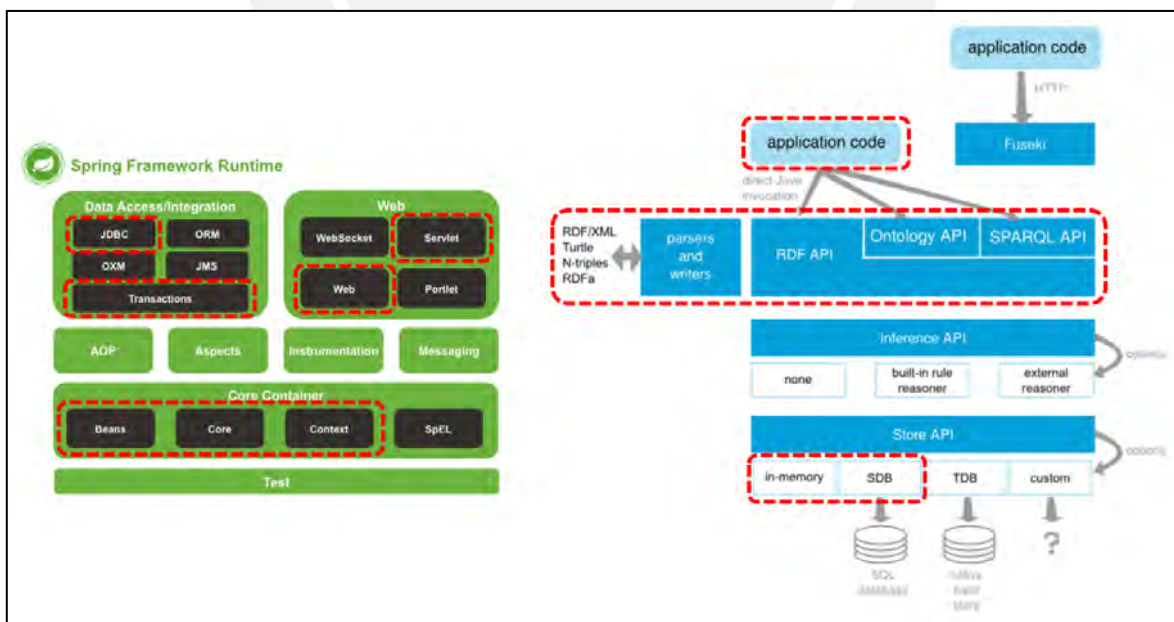


Figura 6.2 Framework Spring y Jena (Johnson et al., 2016)

6.5 Prototipos

Cada una de las vistas (páginas) de la interfaz web tendrá la siguiente distribución:

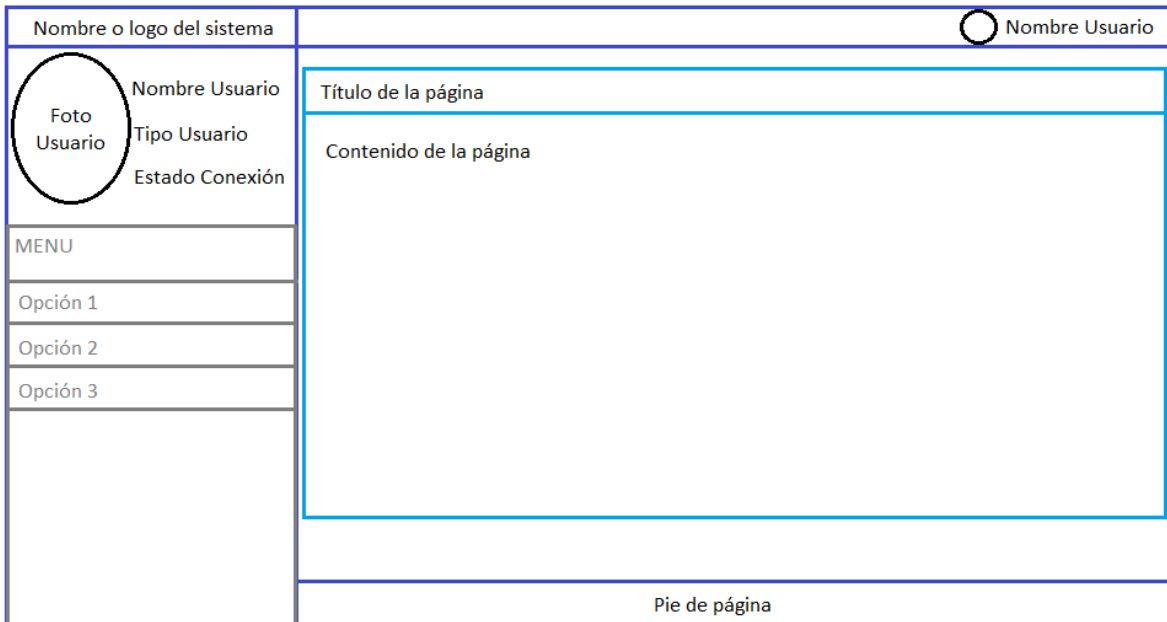
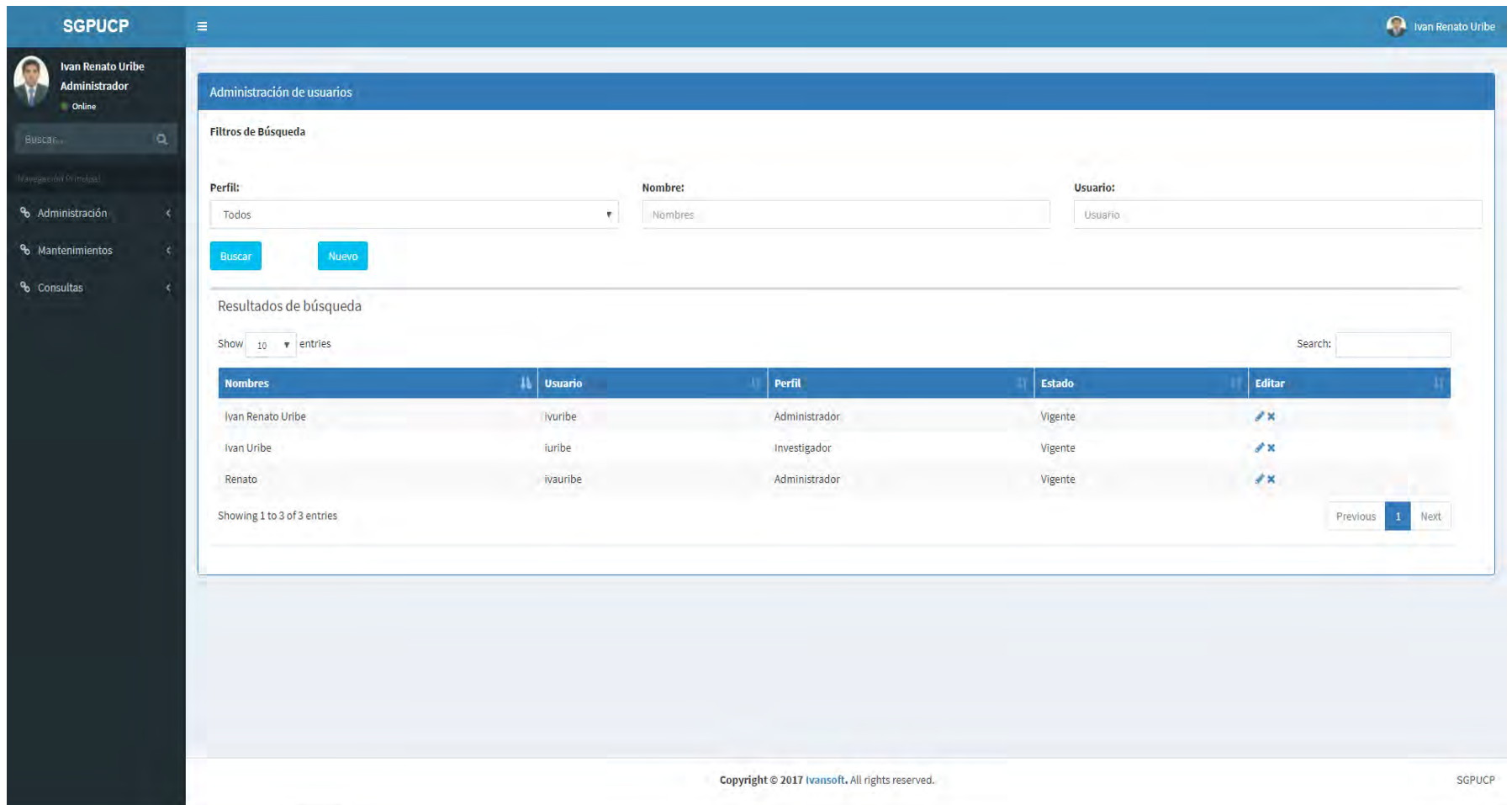


Figura 6.3 Estructura del prototipo

6.5.1 Mantener Usuarios

Módulo de búsqueda de usuarios



The screenshot displays the 'Administración de usuarios' (User Management) module in the SGPUCP system. The interface includes a top navigation bar with the SGPUCP logo and the user's name 'Ivan Renato Uribe'. A left sidebar shows navigation options: 'Administración', 'Mantenimientos', and 'Consultas'. The main content area features search filters for 'Perfil' (set to 'Todos'), 'Nombre' (set to 'Nombres'), and 'Usuario' (set to 'Usuario'). Below the filters are 'Buscar' and 'Nuevo' buttons. The search results section shows 'Resultados de búsqueda' with a 'Show 10 entries' dropdown and a search input field. A table lists three users with columns for 'Nombres', 'Usuario', 'Perfil', 'Estado', and 'Editar'. The table data is as follows:

Nombres	Usuario	Perfil	Estado	Editar
Ivan Renato Uribe	ivuribe	Administrador	Vigente	✎ ✕
Ivan Uribe	iuribe	Investigador	Vigente	✎ ✕
Renato	ivauribe	Administrador	Vigente	✎ ✕

At the bottom of the table, it indicates 'Showing 1 to 3 of 3 entries' and includes 'Previous', '1', and 'Next' navigation buttons. The footer contains the copyright notice 'Copyright © 2017 Ivansoft. All rights reserved.' and the SGPUCP logo.

Módulo de creación o actualización de usuarios

SGPUCP Ivan Renato Uribe

Ivan Renato Uribe
Administrador
Online

Buscar

- Administración
- Mantenimientos
- Consultas

Nuevo Usuario

Datos del usuario

Nombres: **Apellido Paterno:**

Apellido Materno: **Perfil de Usuario:**

Email:

Credenciales de inicio de sesión

Usuario:

Contraseña:

Confirmar Contraseña:

Copyright © 2017 Ivansoft. All rights reserved. SGPUCP

6.5.2 Mantener Autores

Módulo de búsqueda de autores

SGPUCP Ivan Renato Uribe

Ivan Renato Uribe
Administrador
Online

Buscar...

Navegación Principal

- Administración
- Mantenimientos
- Consultas

Administración de Autores

Filtros de Búsqueda

Nombre: **Alias:** **Estado:**

[Buscar](#) [Nuevo](#)

Resultados de búsqueda

Show entries Search:

Nombre	Institución	Editar
Abraham Eliseo Dávila Ramón	Pontificia Universidad Catolica del Peru	✎ ✕
Claudia Zapata	Pontificia Universidad Catolica del Peru	✎ ✕
Hector Andrés Melgar Sasieta	Pontificia Universidad Catolica del Peru	✎ ✕
José Antonio Pow-Sang Portillo	Pontificia Universidad Catolica del Peru	✎ ✕
Luis Alberto Flores García	Pontificia Universidad Catolica del Peru	✎ ✕
Manuel Tupia Anticona	Pontificia Universidad Catolica del Peru	✎ ✕
Maynard Kong	Pontificia Universidad Catolica del Peru	✎ ✕
Rony Cueva Moscoso	Pontificia Universidad Catolica del Peru	✎ ✕

Showing 1 to 8 of 8 entries Previous **1** Next

Copyright © 2017 Ivansoft. All rights reserved. SGPUCP

Módulo de creación o actualización de autores

The screenshot displays the 'Actualizar Autor' (Update Author) interface within the SGPUCP system. The interface is divided into several sections:

- Header:** 'SGPUCP' logo on the left and user profile 'Ivan Renato Uribe' on the right.
- Left Sidebar:** A dark navigation menu with options: 'Administración', 'Mantenimientos', and 'Consultas'.
- Form Section (Datos del Autor):** Contains two input fields: 'Nombre:' with the value 'Hector Andrés Melgar Sasieta' and 'Institución:' with a dropdown menu set to 'Pontificia Universidad Católica del Peru'.
- List Section (Lista de Alias del autor):** Features a search bar, a 'Show 10 entries' dropdown, and a table of aliases. The table has two columns: 'Alias' and 'Editar'.

Alias	Editar
Andrés Melgar	
H. Andrés Melgar S.	
- Footer:** 'Showing 1 to 2 of 2 entries' and pagination controls (Previous, 1, Next). At the bottom, there are 'Guardar' (Save) and 'Cancelar' (Cancel) buttons.
- Page-Footer:** 'Copyright © 2017 Ivansoft. All rights reserved.' and 'SGPUCP' logo.

6.5.3 Consultar Publicaciones

Módulo de búsqueda de publicaciones

The screenshot displays the SGPUCP web application interface. At the top, a blue header bar contains the SGPUCP logo on the left and a user profile for Ivan Renato Uribe on the right. A dark sidebar on the left lists navigation options: Administración, Mantenimientos, and Consultas. The main content area features a search module titled 'Módulo de búsqueda de publicaciones por autor' under the heading 'Publicaciones Académicas y Científicas Peruanas'. This module includes a search bar, a dropdown menu for 'Autor' (currently set to '...Seleccionar'), and a 'Buscar' button. The footer contains the copyright notice 'Copyright © 2017 Ivansoft. All rights reserved.' and the SGPUCP logo.

Módulo de resultados de publicaciones

SGPUCP ☰ Ivan Renato Uribe

 **Ivan Renato Uribe**
Administrador
Online

Buscar... 🔍

Navegación Principal

- Administración <
- Mantenimientos <
- Consultas <

Resultado de búsqueda de Publicaciones Académicas y Científicas Peruanas

Artículos y Publicaciones del Autor: **Manuel Tupia Anticona**

- A Double Relaxation GRASP Algorithm to Solve the Flow-shop Scheduling Problem. +
- Double Relaxation -- GRASP Algorithm for Solve the Table Assignment Problem on Data Base Store Devices. +
- Information security risks in a customer service call center infrastructure. Guidelines for security managers. +
- A GRASP algorithm to solve the problem of dependent tasks scheduling in different machines. +
- GraspKM en la Recuperación de la Estructura de Software. +
- VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento-JIISIC'07, 31 de Enero al 2 de Febrero del 2007, Lima, Perú +
- Two GRASP Metaheuristic for the Capacitated Vehicle Routing Problem Considering Split Delivery and Solving the Three Dimensional Bin Packing Problem. +

[Regresar](#)

Copyright © 2017 **Ivansoft**. All rights reserved. SGPUCP



Ivan Renato Uribe
Administrador
Online

Navegación Principal

Administración

Mantenimientos

Consultas

Resultado de búsqueda de Publicaciones Académicas y Científicas Peruanas

Artículos y Publicaciones del Autor: Manuel Tupia Anticona

A Double Relaxation GRASP Algorithm to Solve the Flow-shop Scheduling Problem.

Información general

Referencia original: A Double Relaxation GRASP Algorithm to Solve the Flow-shop Scheduling Problem.

Autores: <http://dblp.rkbexplorer.com/id/people-d4439379de0128dc189f71ca90058560-a77be4803d9a02640c90f8e7def90072>, <http://dblp.rkbexplorer.com/id/people-9e9e5c090f4a0dca5d0fa81ed9e6cd7b-a77be4803d9a02640c90f8e7def90072>,

Palabras clave: No se encontraron palabras clave.

Abstract: No se encontró el abstract (resumen).

Año de publicación: 2007

Información en formato Turtle

Propiedad	Valor
http://www.aktors.org/ontology/portal#has-date	http://www.aktors.org/ontology/date#2007
http://www.aktors.org/ontology/portal#cites-publication-reference	http://dblp.rkbexplorer.com/id/conf/caine/2007
http://www.aktors.org/ontology/portal#article-of-journal	http://dblp.rkbexplorer.com/id/journal-f68ab0429fd6d192491560fcea08f91d
http://www.aktors.org/ontology/portal#has-author	http://dblp.rkbexplorer.com/id/people-d4439379de0128dc189f71ca90058560-a77be4803d9a02640c90f8e7def90072
http://www.aktors.org/ontology/portal#has-author	http://dblp.rkbexplorer.com/id/people-9e9e5c090f4a0dca5d0fa81ed9e6cd7b-a77be4803d9a02640c90f8e7def90072
http://www.aktors.org/ontology/portal#has-title	A Double Relaxation GRASP Algorithm to Solve the Flow-shop Scheduling Problem.
http://www.w3.org/2002/07/owl#sameAs	http://dblp.l3s.de/d2/r/resource/publications/conf/caine/TupiaR07
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.aktors.org/ontology/portal#Book-Section-Reference
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	637f4372:15d2fa82f0f:7ffc

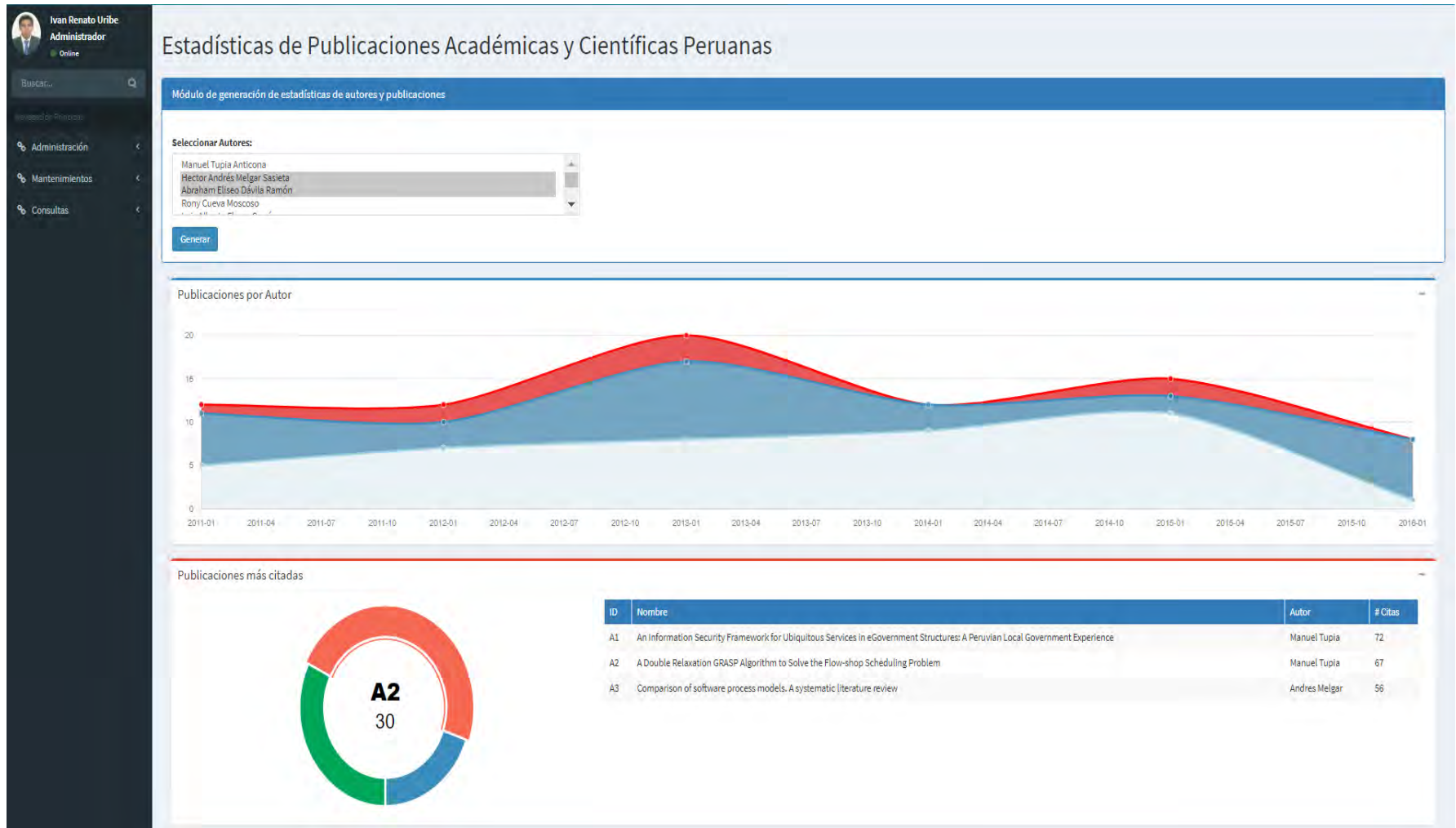
Descargar RDF/XML

Descargar PDF

Double Relaxation -- GRASP Algorithm to Solve the Table Assignment Problem on Data Base Store Devices.

6.5.4 Generar Estadísticas

Módulo de generación de estadísticas



CAPÍTULO 7: Conclusiones

7.1 Escenario de pruebas y resultados

Para comprobar que los algoritmos implementados realizan la búsqueda esperada, se tomará como caso de prueba la búsqueda de las publicaciones del autor Manuel Tupia Anticona en el repositorio DBLP. En este caso, de acuerdo con la *Figura 7.1* y la *Figura 7.2*, el autor presenta un problema de ambigüedad al aparecer en dicho repositorio mediante dos nombres: “Manuel Tupia” y “Manuel Tupia Anticona”. Entre los años 2006 al 2013, cuenta con las siguientes publicaciones indexadas:

- Año 2006:
 - A GRASP algorithm to solve the problem of dependent tasks scheduling in different machines.
- Año 2007:
 - A Double Relaxation GRASP Algorithm to solve the Flow-shop scheduling problem.
 - GraspKM en la Recuperación de la Estructura de Software.
 - VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento.
- Año 2008:
 - Double Relaxation –GRASP Algorithm for Solve the Table Assignment Problem on Data Base Store Services.
- Año 2010:
 - Information Security risks in a customer service call center infrastructure. Guidelines for security managers.
 - Two GRASP Metaheuristic for the Capacitated Vehicle Routing Problem Considering Split Delivery and Solving Three Dimensiona Bin Packing Problem.

dblp.uni-trier.de
Computer Science
Bibliography

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

Universität Trier

Manuel Tupia

Listing of the [DBLP Bibliography Server](#) - [FAQ](#) [Facets and more with CompleteSearch](#)
 Other views (modern): [by type](#) - [by year](#)
 Other mirrors: [Trier II](#) - [Dagstuhl](#)

Ask others: [ACM DL/Guide](#) - [CiteSeerX](#) - [CSB](#) - [MetaPress](#) - [Google](#) - [Bing](#) - [Yahoo](#)

2016	
c4	Manuel Tupia, Mariuxi Bruzza , Flavio Rodriguez : An Information Security Framework for Ubiquitous Services in eGovernment Structures: A Peruvian Local Government Experience. FedCSIS 2016 : 1309-1316
2010	
c3	Manuel Tupia: Information security risks in a customer service call center infrastructure. Guidelines for security managers. NCM 2010 : 262-264
2008	
c2	Rony Cueva , Manuel Tupia: Double Relaxation -- GRASP Algorithm for Solve the Table Assignment Problem on Data Base Store Devices. NCM (2) 2008 : 418-421
2007	
c1	Manuel Tupia, César Ramirez : A Double Relaxation GRASP Algorithm to Solve the Flow-shop Scheduling Problem. CAINE 2007 : 1-4

Figura 7.1 Publicaciones de “Manuel Tupia” (DBLP, s. f.-a)

← ↻ dblp.uni-trier.de/pers/hc/a/Anticona:Manuel_Tupia

Manuel Tupia Anticona

Listing of the [DBLP Bibliography Server](#) - [FAQ](#) [Facets and more with CompleteSearch](#)
 Other views (modern): [by type](#) - [by year](#)
 Other mirrors: [Trier II](#) - [Dagstuhl](#)

Ask others: [ACM DL](#) / [Guide](#) - [CiteSeer^X](#) - [CSB](#) - [MetaPress](#) - [Google](#) - [Bing](#) - [Yahoo](#)

2010	
j1	Joseph Gallart Suarez , Manuel Tupia Anticona: Two GRASP Metaheuristic for the Capacitated Vehicle Routing Problem Considering Split Delivery and Solving the Three Dimensional Bin Packing Problem. AISS 2 (2): 42-50 (2010)
c3	Joseph Gallart Suarez , Manuel Tupia Anticona: Solving the Capacitated Vehicle Routing Problem and the Split Delivery Using GRASP Metaheuristic. IFIP AI 2010 : 243-249
2007	
c2	Erick Vicente , Manuel Tupia Anticona, Luis Rivera : GraspKM en la Recuperación de la Estructura de Software. JIISIC 2007 : 35-42
e1	Maynard Kong , José Antonio Pow-Sang , Manuel Tupia Anticona, Luis Alberto Flores Garcia : VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento-JIISIC'07, 31 de Enero al 2 de Febrero del 2007, Lima, Perú. Facultad de Ciencias e Ingeniería and Departamento de Ingeniería, Pontificia Universidad Católica del Perú 2007 , ISBN 978-9972-2885-1-7
2006	
c1	Manuel Tupia Anticona: A GRASP algorithm to solve the problem of dependent tasks scheduling in different machines. IFIP AI 2006 : 325-334

Figura 7.2 Publicaciones de “Manuel Tupia Anticona” (DBLP, s. f.-b)

Cuando el software ejecuta la búsqueda, se obtiene el siguiente resultado en la interfaz:

The screenshot displays the SGPUCP web interface. At the top left, the logo 'SGPUCP' is visible. On the right side of the top bar, the user 'Ivan Renato Uribe' is logged in. A left sidebar contains a user profile for 'Ivan Renato Uribe, Administrador' and a navigation menu with items: 'Administración', 'Mantenimientos', and 'Consultas'. The main content area is titled 'Resultado de búsqueda de Publicaciones Académicas y Científicas Peruanas'. Below this title, a blue header indicates 'Artículos y Publicaciones del Autor: Manuel Tupia Anticona'. The results are listed in a table with seven entries, each with a plus sign on the right for expansion:

Artículos y Publicaciones del Autor: Manuel Tupia Anticona	
A Double Relaxation GRASP Algorithm to Solve the Flow-shop Scheduling Problem.	+
Double Relaxation -- GRASP Algorithm for Solve the Table Assignment Problem on Data Base Store Devices.	+
Information security risks in a customer service call center infrastructure. Guidelines for security managers.	+
A GRASP algorithm to solve the problem of dependent tasks scheduling in different machines.	+
GraspKM en la Recuperación de la Estructura de Software.	+
VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento-JIISIC'07, 31 de Enero al 2 de Febrero del 2007, Lima, Perú	+
Two GRASP Metaheuristic for the Capacitated Vehicle Routing Problem Considering Split Delivery and Solving the Three Dimensional Bin Packing Problem.	+

At the bottom left of the results area, there is a blue button labeled 'Regresar'.

Figura 7.3 Resultados de publicaciones de “Manuel Tupia Anticona”

Según los resultados obtenidos, las siete publicaciones del autor son encontradas desde el buscador de la interfaz. Un aspecto relevante del módulo de búsqueda de publicaciones es que puede realizar la búsqueda en cualquier dataset que previamente se haya registrado en el Mantenimiento de Datasets. La búsqueda se realiza mediante el Sparql Endpoint asociado al dataset. En caso dicho intento de búsqueda falle (por problemas con el Endpoint) el algoritmo realizará una segunda búsqueda a nivel local en el conjunto de archivos rdf que se hayan registrado para el Dataset. En el caso de DBLP, el Endpoint asociado es <http://dblp.rkbexplorer.com/sparql/>. Dicho Endpoint sólo administra información RDF entre el 2006 y 2013. Por lo tanto, para el autor de prueba su publicación del 2016 no aparece en los resultados de búsqueda. Otro aspecto para resaltar es que el problema de la ambigüedad (dos nombres asociados al mismo autor) no resultó ser un inconveniente para el algoritmo. Este problema se resuelve con el Mantenimiento de Autores, donde es posible especificar una lista de alias para el autor aparte de su nombre completo.

7.2 Observaciones del proyecto

De acuerdo con las pruebas realizadas en los datasets escogidos, se ha podido observar que estos datasets reutilizan ontologías top-level y definen, a su vez, una ontología de dominio que permite manejar el contexto específico. Esta tendencia ha permitido que se establezca un estándar para el manejo de los datos, el cual ha facilitado la necesidad de buscar de forma integral y automática en distintos repositorios, la cual ha sido cubierta por la alternativa de solución del presente proyecto de tesis.

Para la comprobación de resultados se utilizó información conocida de autores peruanos que son profesores de la PUCP, y se verificó que los resultados de búsqueda eran los correctos comparándolos con una búsqueda manual en los repositorios. La potencialidad de la solución está en la capacidad de poder modelar una ontología de acuerdo con cada dataset donde se realizará la búsqueda. A partir de dicha ontología, se crean los métodos de consultas de información mediante SPARQL. Finalmente, con el tratamiento de grafos RDF la información se puede serializar y manipular para su presentación a los distintos usuarios.

Cabe mencionar que para aumentar la capacidad del algoritmo se permite manejar un corpus de palabras claves, términos y entidades relevantes para el contexto. Incluso se puede gestionar manualmente un diccionario de ontologías y datasets conocidos, el cual también es realimentado cuando el algoritmo detecta que las clases y predicados o el dataset es uno nuevo y no se encuentra en la base de conocimiento almacenada.

7.3 Conclusiones

En esta sección se presentan las conclusiones sobre cada uno de los objetivos específicos y los resultados esperados, así como las conclusiones generales sobre las pruebas y resultados alcanzados.

7.3.1 Conclusiones sobre los objetivos y resultados esperados

El presente trabajo tiene como objetivo general la implementación de un software para la búsqueda de publicaciones científicas mediante Datos Enlazados. En tal sentido, cada uno de los objetivos específicos planteados ha sido abordado por la solución implementada, de tal manera que se satisfacen los resultados esperados para cada uno de dichos objetivos. Las conclusiones son las siguientes:

1. Para el OE1 se logró implementar una ontología base que posee un corpus base de palabras clave que establece un contexto para la búsqueda de publicaciones y autores. Esta ontología puede ser adaptada en tiempo de ejecución según el dataset que se consulta. Eso quiere decir que la ontología base es ampliada o modificada mediante un algoritmo que tiene la capacidad de detectar en el dataset nuevas palabras claves relevantes para el contexto. De esta forma, la ontología posee la estructura necesaria que permite la gestión de la información.
2. Para el OE2 se implementaron los algoritmos que, haciendo uso de la ontología adaptativa, permiten la generación de las consultas en SPARQL. La solución sigue una serie de procedimientos para poder diseñar, en tiempo de ejecución, la consulta en SPARQL. Primero, reconoce las propiedades (datos) que gestiona el dataset; luego, recupera las URI que corresponden a un mismo autor (resolviendo el problema de ambigüedad). Para cada URI del autor se obtienen las URI de sus publicaciones. Finalmente, se diseña el SPARQL para obtener las tuplas de cada publicación. En el escenario de pruebas se observa que se obtienen los metadatos relevantes sobre cada publicación, se puede saber principalmente: año de publicación, coautores, palabras clave, resumen (abstract), otras publicaciones referenciadas, link de la referencia original de la publicación y link de referencia para descarga.
3. Para el OE3 se establece el algoritmo que permita la transformación de los metadatos obtenidos a un formato legible para el usuario final. Este algoritmo hace uso de la ontología generada para gestionar y almacenar las tuplas encontradas. Los metadatos se reestructuran en un formato de cabecera (información general de la publicación: autores, palabras clave, abstract y año de

publicación) y detalle (el resto de las propiedades). Toda la estructura de los metadatos es almacenada en un archivo RDF/XML. Este formato se evidencia a nivel de usuario mediante los resultados esperados del OE4.

4. Para el OE4 se logró la implementación de los módulos de búsqueda de publicaciones y de generación de estadísticas de publicaciones por autor. El escenario de pruebas permite comprobar que el módulo de búsquedas y el de estadísticas, según las precondiciones mencionadas en el punto 7.1, emiten los mismos resultados si es que se hiciera la búsqueda de forma manual en el buscador de DBLP. Asimismo, para cada publicación obtenida se obtienen los metadatos descritos en el OE3. La búsqueda es eficiente (en el orden de los ms.) debido a que los repositorios cuentan con un EndPoint que permite la consulta directa con SPARQL. Estos repositorios con Datos Enlazados permiten la búsqueda integral de los metadatos de las publicaciones, e incluso permiten referenciar a las páginas originales de los artículos de donde se puede descargar el documento completo.

7.3.2 Conclusiones Generales

Luego del desarrollo del presente proyecto y con los resultados obtenidos se concluye, de manera general, que:

- Para consultar un dataset es importante conocer y entender la especificación ontológica que da estructura a los datos. Al tener conocimiento de dicha ontología, es posible realizar procesos automatizados de extracción de información.
- El diseño de una ontología estática sería limitado ya que siempre estará sujeto a modificaciones o ampliaciones de tal modo que pueda gestionar mayor cantidad de datos o entidades relevantes para el dominio. En tal sentido, la lógica del algoritmo adaptativo permite a la solución ser escalable y permite soportar los cambios y la constante evolución.
- Se puede afirmar que el presente proyecto de fin de carrera presenta una alternativa de solución al problema planteado. Se permite la búsqueda de información de autores y publicaciones en distintos repositorios digitales con resultados de búsqueda óptimos y esperados de acuerdo con una comparación de búsqueda manual.

7.4 Recomendaciones

Se proponen los siguientes trabajos futuros que permitirán ampliar y mejorar el presente proyecto de fin de carrera:

- Implementar un algoritmo de desambiguación y procesamiento de lenguaje natural, de tal forma que se puedan realizar búsquedas generales sobre las publicaciones o autores mediante un buscador de conceptos enlazados.
- Ampliar la plataforma con un repositorio digital propio con sus propios procedimientos de publicación de investigaciones y artículos donde se podrán indexar las publicaciones de los investigadores.
- Implementar un SPARQL Endpoint para brindar una interfaz de consultas para usuarios expertos que busquen información especializada.
- Implementar una interfaz interactiva para la construcción de ontologías a nivel de usuario que permita almacenar información y la generación de documentos RDF/XML, Turtle y otros estándares existentes.
- Implementar un generador de consultas de SPARQL para usuarios inexpertos que permita, a través de una interfaz interactiva, la manipulación de las clases y propiedades de un dataset para generar búsquedas.

Bibliografía

- | IOS Semantic Web Journal. (s. f.). Recuperado 7 de septiembre de 2016, a partir de <http://semantic-web-journal.com/sejp/page/node/900>
- Acerca de | Portal de datos abiertos. (s. f.). Recuperado 9 de mayo de 2018, a partir de <https://data.europa.eu/euodp/es/about>
- Albert-Ludwigs-Universität Freiburg. (2009). DBIS - Datenbanken und Informationssysteme. Recuperado 27 de septiembre de 2016, a partir de <http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B/data.php>
- Ávila Barrientos, E. (2016). Datos enlazados y bibliotecas digitales académicas: una alternativa para el apoyo a la investigación en el entorno digital. Recuperado a partir de <http://repositorio.pucp.edu.pe/index//handle/123456789/52628>
- Berners-Lee, T. (2006). Linked Data - Design Issues. Recuperado 29 de mayo de 2018, a partir de <https://www.w3.org/DesignIssues/LinkedData.html>
- Berners-Lee, T. (s. f.). Universal Resource identifiers in WWW - World Wide Web Consortium. Recuperado 6 de septiembre de 2016, a partir de <https://www.w3.org/Addressing/URL/uri-spec.html>
- Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data - The story so far. *International Journal on Semantic Web and Information Systems*, 5(3), 1-22. <https://doi.org/10.4018/jswis.2009081901>
- Blakley, B. (2010). citability / Citable Documents Specification. Recuperado 26 de septiembre de 2016, a partir de <http://citability.pbworks.com/w/page/7506673/Citable%20Documents%20Specification>
- CONCYTEC. (2014). *Principales indicadores bibliométricos de la actividad científica peruana 2006-2011* (1.ª ed.). CONCYTEC. Recuperado a partir de <https://portal.concytec.gob.pe/index.php/publicaciones/informes/item/48-informe-n-1-principales-indicadores-bibliometricos-de-la-actividad-cientifica-peruana-2006-2011>

- DBLP. (2016). dblp: What is dblp? Recuperado 15 de septiembre de 2016, a partir de <http://dblp.uni-trier.de/faq/What+is+dblp.html>
- DBLP. (s. f.-a). DBLP: Manuel Tupia. Recuperado 14 de julio de 2017, a partir de <http://dblp.uni-trier.de/pers/hc/t/Tupia:Manuel>
- DBLP. (s. f.-b). DBLP: Manuel Tupia Anticon. Recuperado 13 de julio de 2017, a partir de http://dblp.uni-trier.de/pers/hc/a/Anticon:Manuel_Tupia
- DBPEDIA. (s. f.). About: Pedro Paulet. Recuperado 27 de septiembre de 2016, a partir de http://dbpedia.org/page/Pedro_Paulet
- DCMI Metadata Terms. (s. f.). Recuperado 13 de julio de 2017, a partir de <http://dublincore.org/documents/dcmi-terms/>
- Deitel, P., & Deitel, H. (2015). *Java How to Program* (10.^a ed.). Pearson Education. Recuperado a partir de <https://ebooks-it.org/0133807800-ebook.htm>
- FOAF Vocabulary Specification. (s. f.). Recuperado 13 de julio de 2017, a partir de <http://xmlns.com/foaf/spec/>
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5), 907-928. <https://doi.org/10.1006/ijhc.1995.1081>
- Guha, R. V., Brickley, D., & Macbeth, S. (2016). Schema.Org: Evolution of Structured Data on the Web. *Commun. ACM*, 59(2), 44–51. <https://doi.org/10.1145/2844544>
- Guía Breve de Web Semántica. (s. f.). Recuperado 6 de septiembre de 2016, a partir de <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>
- Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., ... Davison, D. (2016). Spring Framework Reference Documentation. Recuperado 4 de octubre de 2016, a partir de <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/>
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Software Engineering Group, Department of Computer Science, Keele University and Empirical Software*

Engineering National ICT Australia Ltd, Keele University Technical Report TR/SE-0401,
33.

Kitchenham, B. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering* (Software Engineering Group, Department of Computer Science, Keele University and Empirical Software Engineering National ICT Australia Ltd, EBSE Technical Report, EBSE-2007-01). Kitchenham.

Linked Open Vocabularies (LOV). (s. f.). Recuperado 13 de julio de 2017, a partir de
<http://lov.okfn.org/dataset/lov/vocabs/akt>

LinkedData - W3C Wiki. (s. f.). Recuperado 6 de septiembre de 2016, a partir de
<https://www.w3.org/wiki/LinkedData>

Manual de uso de ALICIA. (s. f.). Recuperado 30 de abril de 2018, a partir de
<https://sites.google.com/a/concytec.gob.pe/alicia/?pli=1>

Mark, P., & Helen, R. (2006). *Systematic Reviews in the Social Sciences*. Blackwell Publishing Ltd.

Mata, J., Crespo, M., & Maña, M. J. (2011). Estudio del uso de Ontologías para la Expansión de Consultas en Recuperación de Imágenes en el Dominio Biomédico. *Procesamiento del Lenguaje Natural*, 47(0), 39-46.

Mateus, S. P., Ruiz, M. A., & Plaza, J. E. G. (2015). Lenguajes de recuperación de información sobre la web semántica. *REVISTA POLITÉCNICA*, 5(8), 39-46.

Melgar Sasieta, A. (2017). Situación de las publicaciones científicas en el Perú y su impacto local e internacional. *Universidad Peruana de Ciencias Aplicadas (UPC)*. Recuperado a partir de
<https://repositorioacademico.upc.edu.pe/handle/10757/622408>

Méndez Rodríguez, E. (1999). RDF: UN MODELO DE METADATOS FLEXIBLE PARA LAS BIBLIOTECAS DIGITALES DEL PRÓXIMO MILENIO, 11.

Méndez Rodríguez, E. M. (2001). Metadatos y recuperación de información: estándares, problemas y aplicabilidad en bibliotecas digitales.

- Miao, Q., Meng, Y., Fang, L., Nishino, F., & Igata, N. (2015). Link scientific publications using linked data. En *2015 IEEE International Conference on Semantic Computing (ICSC)* (pp. 268-271). <https://doi.org/10.1109/ICOSC.2015.7050818>
- Noy, N. F., & McGuinness, D. L. (s. f.). *Ontology Development 101: A Guide to Creating Your First Ontology*. *Stanford Medical Informatics*.
- OWL Web Ontology Language Overview. (s. f.). Recuperado 24 de octubre de 2016, a partir de <https://www.w3.org/TR/owl-features/>
- Papadakis, I., Kyprianos, K., & Stefanidakis, M. (2015). Linked data URIs and libraries: The story so far. *D-Lib Magazine*, 21(5-6), 1. <https://doi.org/10.1045/may2015-papadakis>
- Pascal Hitzler, & Krzysztof Janowicz. (s. f.). About the Semantic Web journal (by IOS Press) | www.semantic-web-journal.net. Recuperado 7 de septiembre de 2016, a partir de <http://www.semantic-web-journal.net/>
- Peis, E., Herrera-Viedma, E., Hassan-Montero, Y., & Herrera, J. C. (2003). Ontologías, metadatos y agentes: recuperación «semántica» de la información. Recuperado a partir de <http://digibug.ugr.es/handle/10481/1206>
- Peset, F., Ferrer-Sapena, A., & Subirats-Coll, I. (2011). Open data y Linked open data: su impacto en el área de bibliotecas y documentación. *El Profesional de la Información*, 20(2), 165-174. <https://doi.org/10.3145/epi.2011.mar.06>
- Piedra, N., Chicaiza, J., Quichimbo, P., Cadme, E., Vargas, J. L., Saquicela, V., ... Tovar, E. (2014). Una aproximación basada en linked data para la integración de repositorios digitales abiertos latinoamericanos. En *ResearchGate*. Recuperado a partir de https://www.researchgate.net/publication/284162984_Una_aproximacion_basada_en_linked_data_para_la_integracion_de_repositorios_digitales_abiertos_latinoamericanos
- Piñero López, J. M. (1972). *El análisis estadístico y sociométrico de la literatura científica*. Recuperado a partir de <https://dialnet.unirioja.es/servlet/libro?codigo=622464>
- protégé. (s. f.). Recuperado 12 de julio de 2017, a partir de <http://protege.stanford.edu/about.php>

- RDF 1.1 Turtle. (s. f.). Recuperado 7 de septiembre de 2016, a partir de <https://www.w3.org/TR/turtle/>
- RDF Schema 1.1. (s. f.). Recuperado 13 de julio de 2017, a partir de <https://www.w3.org/TR/rdf-schema/>
- Requisitos para adherirse al repositorio Nacional - Manual de uso de ALICIA. (s. f.). Recuperado 30 de abril de 2018, a partir de <https://sites.google.com/a/concytec.gob.pe/alicia/requisitos-para-adherirse-al-repositorio-nacional>
- Resource Description Framework (RDF) Model and Syntax Specification. (s. f.). Recuperado 12 de julio de 2017, a partir de <https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- Ríos-Hilario, A., Martín-Campo, D., & Ferreras-Fernández, T. (2012). Linked data y linked open data: su implantación en una biblioteca digital. El caso de Europeana. *El Profesional de la Información*, 21(3), 292-297. <https://doi.org/10.3145/epi.2012.may.10>
- SCielo Perú. (2016). SciELO - Scientific Electronic Library Online. Recuperado 14 de septiembre de 2016, a partir de <http://www.scielo.org.pe/>
- SCImago Journal. (2016). SJR - International Science Ranking. Recuperado 14 de septiembre de 2016, a partir de <http://www.scimagojr.com/countryrank.php>
- Shen, Z., Li, J., & Han, F. (2015). Opensdb: Research on The Application of Linked Data in Scientific Databases. *Data Science Journal*, 14(0), 4. <https://doi.org/10.5334/dsj-2015-004>
- SPARQL. (2015, diciembre 9). En *Wikipedia, la enciclopedia libre*. Recuperado a partir de <https://es.wikipedia.org/w/index.php?title=SPARQL&oldid=87677514>
- StarUML 2 Documentation — StarUML 2.0.0 documentation. (s. f.). Recuperado 12 de julio de 2017, a partir de <http://docs.staruml.io/en/latest/index.html>
- Takahiro Komamizu, Toshiyuki Amagasa, & Hiroyuki Kitagawa. (2016). H-SPOOL: A SPARQL-based ETL framework for OLAP over linked data with dimension hierarchy extraction. *International Journal of Web Information Systems*, 12(3), 359-378. <https://doi.org/10.1108/IJWIS-03-2016-0014>

- The Linking Open Data cloud diagram. (s. f.). Recuperado 6 de septiembre de 2016, a partir de <http://lod-cloud.net/>
- Valencia Castillo, E. (s. f.). *Recuperación y organización de la información a través de RDF usando SPARQL*. Universidad Pontificia de Salamanca–Campus de Madrid. gomez. files. wordpress. com/2008/09/informe-sparql. doc Sitios Web [14] Apache Jena-SPARQL Tutorial. <http://jena.apache.org/tutorials/sparql.html>.
- W3C. (2012). OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). Recuperado 13 de julio de 2017, a partir de <https://www.w3.org/TR/owl2-syntax/>
- W3C. (2013). SPARQL 1.1 Query Language. Recuperado 7 de septiembre de 2016, a partir de <https://www.w3.org/TR/sparql11-query/#specDataset>
- W3C. (2014). RDF 1.1 Primer. Recuperado 6 de septiembre de 2016, a partir de <https://www.w3.org/TR/rdf11-primer/>
- W3C. (s. f.). Guía Breve de Linked Data. Recuperado 6 de septiembre de 2016, a partir de <http://www.w3c.es/Divulgacion/GuiasBreves/LinkedData>
- Wiljes, C., Jahn, N., Lier, F., Paul-Stueve, T., Vompras, J., Pietsch, C., & Cimiano, P. (2013). Towards Linked Research Data: An Institutional Approach. En *3rd Workshop on Semantic Publishing (SePublica)*. Recuperado a partir de <https://pub.uni-bielefeld.de/publication/2580621>