



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN

**Servicios Web para Visualización de Datos Enlazados. Piloto datos
enlazados de recursos abiertos.**

TRABAJO DE TITULACIÓN

AUTOR: Yaguachi Pereira, Jonathan David

DIRECTOR: Piedra Pullaguari, Nelson Oswaldo, Ing.

LOJA – ECUADOR

2016



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Septiembre, 2016

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ingeniero.

Nelson Oswaldo Piedra Pullaguari.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Servicios Web para Visualización de Datos Enlazados. Piloto datos enlazados de recursos abiertos realizado por Yaguachi Pereira, Jonathan David, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, marzo de 2016

f).....

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Yaguachi Pereira Jonathan David declaro ser autor del presente trabajo de titulación: Servicios Web para Visualización de Datos Enlazados. Piloto datos enlazados de recursos abiertos, de la Titulación Sistemas Informáticos y Computación, siendo Ing. Piedra Pullaguari Nelson Oswaldo director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f).....

Autor: Yaguachi Pereira Jonathan David.

Cedula: 1104620800.

DEDICATORIA

A mis padres Angel y Almida quienes siempre estuvieron con migo a lo largo de este camino brindándome de su apoyo, comprensión y optimismo especialmente en los momentos difíciles.

A mis hermanos Nixon y Lady, por las experiencias y enseñanzas que juntos hemos vivido, la motivación y el apoyo que siempre recibo de ellos.

A todas esas personas que con su apoyo estuvieron ahí en el transcurso de mi vida universitaria, a los amigos cuya motivación fue un respaldo fundamental en la culminación del presente trabajo.

Jonathan Yaguachi.

AGRADECIMIENTO

A Dios por regalarme la vida, las fuerzas y salud necesaria para llegar hasta este punto de mi vida profesional, por las muchas bendiciones recibidas de su parte y por permitirme culminar una etapa importante en mi formación académica.

A mis padres por su ayuda y comprensión brindada todo el tiempo, por sus consejos que siempre estuvieron ahí para apoyarme, sin sus consejos y aliento, este objetivo quizá no se hubiera cumplido.

A la UTPL por abrirme las puertas de la institución y capacitarme, a mis maestros que gracias a sus conocimientos impartidos han permitido formar personas y prepararlas para el futuro.

Al Ing. Nelson Piedra, director de tesis, quien con sus conocimientos contribuyó de gran manera con el presente trabajo, además a los ingenieros Jorge López, Rene Elizalde miembros del tribunal del presente trabajo, y a todas las personas que de una u otra forma me brindaron su apoyo en la elaboración de este trabajo de investigación.

Jonathan Yaguachi.

ÍNDICE DE CONTENIDOS

CARATULA.....	i
APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS.....	xi
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
Objetivos	3
General.....	3
Específicos.....	3
CAPÍTULO I: ESTADO DEL ARTE.....	5
1.1. El Problema.....	6
1.2. Solución Propuesta.....	6
1.3. Servicios Web	8
1.3.1. WSDL.....	9
1.3.2. SOAP.....	9
1.3.3. UDDI.....	10
1.4. REST (Representational State Transfer)	11
1.4.1. Identificación de Recursos.....	11
1.4.2. Manipulación de Recursos.....	11
1.4.3. Mensajes Autodescriptivos.....	11
1.4.4. Hipermedia como el motor de las aplicaciones.....	12
1.5. Jersey.....	12
Componentes.....	12
1.6. Jackson	12
1.7. JSON.....	13
1.7.1. Estructura JSON.....	13
1.8. Linked Data	13
1.9. Big Data	14

1.9.1.	Volumen.....	14
1.9.2.	Variedad.....	14
1.9.3.	Velocidad.....	15
1.9.4.	Veracidad.....	15
1.10.	Virtuoso	16
1.11.	SPARQL.....	16
1.12.	OER.....	16
1.13.	Visualización de Datos.....	17
1.13.1.	¿Por qué se visualiza?	18
1.14.	Patrón de Visualización.....	19
1.15.	Exhibit.....	20
1.16.	JavaScript InfoVis Toolkit.....	20
1.17.	Arbor.js	20
1.18.	D3.js.....	20
1.19.	SigmaJS	20
1.20.	Trabajos relacionados.....	24
CAPÍTULO II: DESCRIPCIÓN DE DATOS.....		26
2.1	Introducción.....	27
2.2	Datos a trabajar	27
2.2.1	Sujeto.....	28
2.2.2	Predicado.....	28
2.2.3	Objeto.	28
2.3	Vocabularios	28
2.3.1	RDF.....	29
2.3.2	Rdf-schema.	30
2.3.3	SKOS.....	31
2.3.4	Dublin Core.....	32
2.3.5	FOAF.	33
2.4	Clasificación de los Datos.....	34
2.5	Representación Grafica de los Vocabularios	35
CAPÍTULO III: SERVICIOS WEB		44
3.1	Introducción.....	45
3.2	Arquitectura.....	45
3.2.1	Virtuoso.....	46
3.2.2	Consulta.....	47

3.2.3	Parser.....	63
3.2.4	Cliente.....	71
3.2.5	Preguntas que Resuelven Estas Aproximaciones.....	73
DISCUSIÓN.....		76
Descripción de Datos.....		76
Patrones de Visualización		76
Servicios Web		77
Servicio de Consulta.....		77
Servicio de Parser.....		78
Cliente Web.....		78
CONCLUSIONES.....		80
RECOMENDACIONES		82
BIBLIOGRAFÍA.....		83
ANEXOS.....		86
ANEXO 1. DIAGRAMA DE SECUENCIA APLICACION DE CONVERSION.....		87
ANEXO 2. PROCESO DE CONVERSIÓN		88
ANEXO 3. MANUAL DE APLICACIÓN		89
Introducción.....		89
Descripción.....		89
Elementos Principales.....		90
Gráficas		95

ÍNDICE DE FIGURAS

Figura 1. Complejidad Cognitiva	7
Figura 2. Flujo de Trabajo Web Service	8
Figura 3: Estructura SOAP.....	10
Figura 4: Patrones de percepción humana.....	19
Figura 5. Representación Gráfica de una Tripleta	28
Figura 6. Tripleta, relación Autor – Publicación.....	29
Figura 7. Grafo Vocabularios Tesis	35
Figura 8. Grafo Vocabularios Maestría.....	36
Figura 9. Grafo Vocabularios Video	37
Figura 10. Grafo Vocabularios Libro.	38
Figura 11. Grafo Vocabularios Reporte.....	39
Figura 12. Grafo Vocabularios Documento.....	40
Figura 13. Grafo Vocabularios Presentación.....	41
Figura 14. Grafo Vocabularios Artículo.....	42
Figura 15. Grafo Vocabularios Recurso de Aprendizaje.....	43
Figura 16. Arquitectura Servicios Web	46
Figura 17. Archivo queries.csv	50
Figura 18. Ejemplo Consulta Sparql	51
Figura 19. Ejemplo Consulta Sparql, Reemplazo de parámetros.....	52
Figura 20. Resultado EndPoint Virtuoso	53
Figura 21. Conversión de JSON	54
Figura 22. JSON de Obtención de Institución.	55
Figura 23. JSON de Obtención de Tipos de Recursos.	56
Figura 24. Obtención de OERs.	58
Figura 25. Estructura JSON General.....	59
Figura 26. Descripción Nodos.	60
Figura 27. Descripción Relaciones (Links)	61
Figura 28. Archivo de Equivalencias.....	62
Figura 29. Cambio de Etiqueta JSON	62
Figura 30. JSON de Obtención del Número de Oers por Tipo de Recurso.	63
Figura 31. Conversión entre JSON.....	65
Figura 32. Listado de publicaciones generado para la gráfica tipo 3.....	70
Figura 33. JSON de Obtención del Número de Oers por Tipo de Recurso, Parser.	71
Figura 34. Autores para una determinada Institución.	73
Figura 35. Publicaciones que han realizado un los Autores de un determinada Institución...74	
Figura 36. Recursos que ha realizado un determinado Autor.....	74
Figura 37. Número de recursos por Tipo de Recurso.	75
Figura 38. Diagrama de Secuencia – Conversor.....	87
Figura 39. Frontal de la Aplicación.	90
Figura 40. Menú de la Aplicación.....	90
Figura 41. Campos Para el Filtro de Datos.....	90
Figura 42. Campos Para el Filtro de Datos.....	91
Figura 43. Mostrar/Ocultar Lista de Nodos.	91

Figura 44. Lista de Nodos.	92
Figura 45. Búsqueda de Nodos.....	92
Figura 46. Pop up, Información de Nodo.	93
Figura 47. Información de Nodo.....	94
Figura 48. Mostrar / Ocultar Información de Nodo.....	94
Figura 49. Mostrar/Ocultar panel de Búsqueda Avanzada.....	94
Figura 50. Búsqueda Avanzada.	95
Figura 51. Force Direct.	96
Figura 52. Tree.	97
Figura 53. Reddit.	98

ÍNDICE DE TABLAS

Tabla 1. Formato de parámetros WSDL	9
Tabla 2. Métodos HTTP utilizados en REST	12
Tabla 3. Formato JSON de las herramientas seleccionadas	21
Tabla 4. Ejemplo Tripleta	28
Tabla 5. Términos RDF	29
Tabla 6. Términos RDFS.....	30
Tabla 7. Términos SKOS	31
Tabla 8. Términos Dublin Core	32
Tabla 9. Términos FOAF	33
Tabla 10. Tipos de Datos Virtuoso.....	34
Tabla 11. Numero de recursos Serendipity OAR	47
Tabla 12. Especificación de Caso de Uso, Consulta de Datos	48
Tabla 13. Consultas Sparql	49
Tabla 14. Reemplazo de Parámetros Consultas Sparql.	51
Tabla 15. Descripción Obtención de Instituciones.	54
Tabla 16. Descripción de Obtención de Tipos de Recursos.	55
Tabla 17. Obtención de OERs.	56
Tabla 18. Descripción Obtención del Número de Oers por Tipo de Recurso.....	62
Tabla 19. Especificación de Caso de Uso, Servicio Web Parser.....	64
Tabla 20. Código de Grafica y grafica a la cual representan	66
Tabla 21. Obtener Listado de OERs.....	68
Tabla 22. Descripción Obtención del Número de Oers por Tipo de Recurso.....	70
Tabla 23. Especificación de Caso de Uso, Cliente	72

RESUMEN

En el presente trabajo de investigación se describe la creación de una aplicación web para la visualización de datos enlazados la misma que será de mucha utilidad a la hora de analizar e interpretar grandes cantidades de información; los datos con los que se trabajó son recursos educativos abiertos publicados por la Universidad Técnica Particular de Loja y se encuentran disponibles en Virtuoso en formato tripleta (sujeto, predicado y objeto) cumpliendo con los principios de Linked Data.

Las herramientas usadas para la visualización de datos son herramientas basadas en la web como HTML, CSS y JavaScript, por esta razón, para el intercambio de la información y el manejo de los datos, se crearon servicios web REST para facilitar este proceso y garantizar la interoperabilidad entre los diferentes módulos y capas de información de la aplicación, además de facilitar la comunicación en un formato común y simple como JSON.

PALABRAS CLAVES: Visualización de Datos, Linked Data, RDF, Servicios Web.

ABSTRACT

The present work describe the creation of a web application for linked data visualization, it will be very useful when analyzing and interpreting large amounts of information; the data with which I worked are open educational resources published by the Universidad Técnica Particular de Loja and are available in triplet format in Virtuoso (subject, predicate and object) in compliance with the principles of Linked Data.

The tools used for data visualization are based on the web as HTML, CSS and JavaScript, therefore, to exchange information and data management, REST web services have been created to facilitate this process and ensure interoperability between different modules and layers information of the application and facilitates communication in a common and simple format like JSON.

KEYWORDS: Data Visualization, Linked Data, RDF, Web Services.

INTRODUCCIÓN

En la actualidad vivimos en una sociedad sobrecargada de información, un exceso de datos que gracias al proceso operativo de los sistemas de información que día a día alimentan grandes bases, estos con el tiempo han crecido tanto que cada vez es más complicado el análisis rápido y fácil de los datos así como su interpretación.

La Visualización de Datos pretende ser la solución a este problema ya que una gran parte de la percepción de información que se recibe es visual (Few, 2013), es así que visualizando los datos se puede reconocer diferentes patrones y conexiones importantes que clasificándolos de la forma correcta se puede darles un significado valioso ya que logran explicar el por qué sucedieron ciertas cosas o simplemente se puede centrar en la información más relevante. (McCandless, 2010)

A partir de esto se pretende hacer la búsqueda de patrones de visualización en formato JSON en las herramientas de visualización disponibles en la Web de entre las cuales se destacan herramientas como D3, Infovis, Arbor, entre otras.

Lo que se plantea es crear un servicio web REST para que realice la gestión de consulta de datos RDF provenientes de una base de datos Virtuoso y generar la estructura JSON necesaria para que el Cliente pueda hacer uso de estos y aplicarlo a una herramienta de visualización.

Los datos a trabajar son recursos educativos abiertos OERs publicados por la Universidad Técnica Particular de Loja, trabajos de fin de titulación de pregrado y postgrado además de otros recursos.

El objetivo es poder determinar visualmente cierto comportamiento que presentan los recursos educativos abiertos, poder observar los patrones y conexiones importantes y encontrar un sentido en ellos, para poder percibir fácilmente el conocimiento que existe detrás de ellos.

Objetivos

General.

- Desarrollar Servicios Web para visualizar datos enlazados de recursos educativos abiertos publicados por la Universidad Técnica Particular de Loja.

Específicos.

- Analizar y describir la estructura de los datos (OERs) almacenados en el EndPoint Serendipity OAR.
- Diseñar patrones de visualización para estructurar los datos con los que se trabajara.

- Desarrollar un Servicio Web para obtener los datos desde virtuoso y estructurarlos al formato JSON a devolver.
- Desarrollar un Servicio Web que permita el intercambio de una estructura JSON a otra, según las necesidades que exija el patrón de visualización.
- Desarrollar un Cliente Web utilizando herramientas disponibles en línea para graficar los datos obtenidos desde el Servicio Web.

CAPÍTULO I: ESTADO DEL ARTE

1.1 El Problema

Los recursos educativos abiertos (OERs) en la actualidad se han convertido en una herramienta muy útil para la educación y el aprendizaje autodidacta ya que presta algunas facilidades y ventajas, dentro las cuales se puede decir que permiten el intercambio de conocimiento, el estudio a distancia, y lo mejor de todo es que son completamente gratuitos haciendo de la educación cada vez más asequible a muchas personas.

Las tecnologías de la información han jugado un papel muy importante en este ámbito ya que han hecho posible la extensión de estos recursos a una gran cantidad de sitios en distintos lugares del planeta.

Es así como en la actualidad muchas instituciones educativas, gubernamentales y organizaciones sin fines de lucro se han unido a esta iniciativa para apoyar al conocimiento y al enriquecimiento intelectual, con el tiempo cada vez son más las instituciones que se unen a este tipo de iniciativas creando, gestionando y compartiendo muchos de sus recursos educativos.

El problema surge cuando se pretende registrar estos cursos para su estudio y análisis; esto para poder recomendar alguno en base a ciertos criterios, gustos e interés.

Se tiene una Base de Datos con la información de recursos educativos impartidos en cientos de universidades en el mundo, el volumen de esta información es tan grande que a menudo se hace muy complicado el procesamiento de esta información.

Tanto como para los recursos tecnológicos como para las personas encargadas de realizar el análisis y estudio de estos recursos, se ven afectados por este problema ya que el procesamiento normal con una base de datos sobrecargada de información lleva mucho tiempo de respuesta usando la tecnología tradicional.

1.2 Solución Propuesta

Frente a la gran información de recursos educativos abiertos que se tiene, la visualización de datos pretende ser una solución ya que de acuerdo a Few el cerebro humano de forma natural encuentra patrones en lo que ve y los relaciona con algo que ya se ha experimentado en ocasiones anteriores (Few, 2013), de esta forma se puede percibir diferentes patrones de visualización que ayuden a identificar el comportamiento de lo que se analiza y también en la toma de decisiones.

La idea es que, de los datos se pueden obtener un entendimiento y criterio de sabiduría y para esto se debe pasar por el proceso de complejidad que se ilustra en la Figura 1. Complejidad Cognitiva, nace del conjuntos de datos que se obtienen de distintas fuentes de información, continua con la conversión de estos datos en información, este proceso se lo realiza estructurando los datos estableciendo un formato determinado y encontrar en ellos

cierto sentido para el usuario; el conocimiento nace de la asimilación personal con los que cada individuo encuentra al analizar la información y encontrando un objetivo para el uso y aplicación dentro de un área de conocimiento con la cual esta información sería de utilidad.



Figura 1. Complejidad Cognitiva
Fuente: (González Díez, 2013)

La visualización es un factor muy importante en el aprendizaje, ya que es de mucha utilidad a la hora de entender muchos datos complejos que no se podrían comprender de otra forma, a continuación algunas ventajas:

- La visualización de datos ayuda a saltar de los datos a la información, reduciendo el esfuerzo cognitivo.
- Ayuda a la generación de conocimiento e ideas fácilmente.
- Enriquecer el análisis de datos

La información existente ya ha sido previamente analizada y estructurada se encuentra almacenada en el TripleStore Virtuoso, para cual se ha planificado las siguientes actividades puntuales:

- Búsqueda de Patrones de visualización JSON en las herramientas disponibles en la web (D3, Infovis, Arbor).
- Crear un Servicio Web para leer los datos desde el End Point de Virtuoso y devolver los datos en un formato Json en una estructura estándar.
- Crear un Servicio Web para consumir los datos que se encuentran en formato Json en una estructura estándar y exportar los resultados en formato JSON, de acuerdo a la estructura que requiera el patrón.
- Creación de un Cliente Web para consumir el JSON provisto por el Servicio web y visualizar la gráfica seleccionada.

1.3 Servicios Web

Es una tecnología que ayuda a la interoperabilidad de los sistemas y aplicaciones existentes hoy en día, esto permite trabajar sobre aplicaciones y sistemas desarrollados en distintos ámbitos como el sistema operativo, lenguajes de programación y diferentes tecnologías utilizadas.

La forma de cómo se realiza esta comunicación es a través de un intercambio de mensajes para lo cual es necesaria la presencia de entidades claves denominadas agentes; un agente puede ser un artefacto de software o un dispositivo hardware que envían y reciben mensajes (W3C, 2004).

Se requieren como mínimo dos agentes en el intercambio de mensajes, el primero de ellos es el agente que realiza la petición; persona u organización que realiza una solicitud para consumir algún recurso, cuando se envía la petición se especifica que datos se espera recibir a través de un mensaje de envío hacia el agente que responde a la petición. El agente que provee el servicio es el encargado de gestionar internamente estas peticiones y de devolver un mensaje de respuesta con los datos solicitados. Por lo general este proceso siempre inicia con el agente que realiza la petición pero no siempre es necesario hacerlo de esta forma, también se puede realizar el mismo proceso es sentido contrario. Se puede observar la Figura 2. Flujo de Trabajo Web Service, en donde se muestra con una secuencia de pasos el proceso que realiza tanto la entidad que hace la petición así como la entidad que responde a dicha entidad.

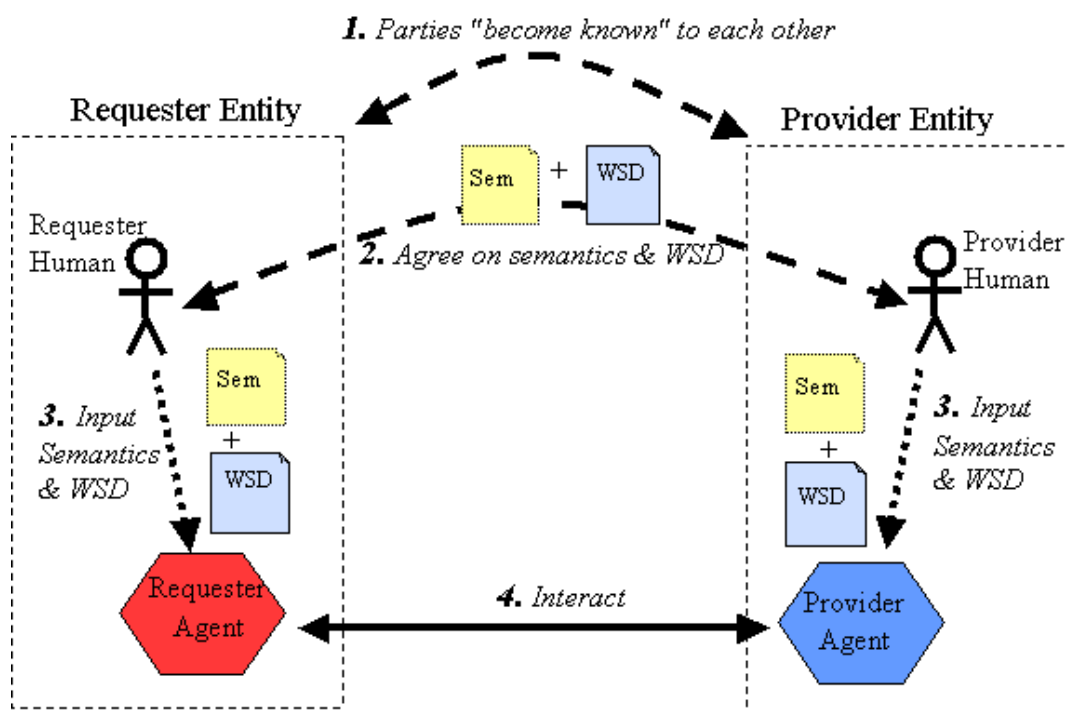


Figura 2. Flujo de Trabajo Web Service

Fuente: (W3C, 2004)

1.3.1 WSDL.

WSDL corresponde a las siglas de Web Services Description Language, está basado en el formato XML y se lo utiliza para describir a los Servicios Web, proveyendo de un formato para la comunicación, dentro de los parámetros descritos por WSDL constan el formato del mensaje, tipos de datos, protocolos de transporte y formatos de serialización de transporte que son usados entre el agente que realiza la petición y el agente que provee el recurso (Wikipedia, 2014). Estos parámetros y su descripción se los puede observar en la Tabla 1. Formato de parámetros WSDL

Tabla 1. Formato de parámetros WSDL

Nombre	Etiqueta XML	Descripción
Datatypes	<types>	Tipos de datos usados en los mensajes
Message	<message>	Definición de los elementos de mensaje
Port Types	<portType>	Definición de las operaciones permitidas y los mensajes intercambiados en el Servicio.
Bindings	<binding>	Protocolos de comunicación usados.
Services	<service>	Conjunto de puertos y dirección.

Fuente: (Wikipedia, 2014)

Elaboración: Jonathan Yaguachi, Junio 2015

1.3.2 SOAP.

SOAP proviene de las siglas en inglés de Simple Object Access Protocol y es un protocolo que ayuda con la gestión de mensajes y la intercomunicación entre dos objetos en dos procesos diferentes y utiliza el formato XML para la comunicación entre ellos, sus características principales son, Extensibilidad, Neutralidad, Independencia (Wikipedia, 2014).

La estructura de SOAP consta de tres partes principales:

1.3.2.1 *Envelope.*

Define el contenido del mensaje, como lo procesara y cuál era la respuesta, es el primero elemento que se visualiza en un mensaje SOAP.

1.3.2.2 *Heder.*

El heder es un parte destinada para almacenar información del mensaje que no es precisamente necesaria, es por eso que el heder es opcional es decir puede como no puede ir dentro de la estructura SOAP, Header se define como un conjunto de reglas que

normalmente están expresadas en metadatos para influir en el procesamiento del contenido del cuerpo (body) SOAP. Si el Heder está presente en un mensaje SOAP debería ser el primer elemento que está dentro del Envelope.

1.3.2.3 Body.

El elemento Body contiene el mensaje (Dato de aplicación) enviado desde un cliente hacia servidor o viceversa (Gustavo, 2008). Contiene básicamente la información que se requiere para consultar en una petición o la información que se desea en el caso de una respuesta.

En la Figura 3: Estructura SOAP ilustra gráficamente cual es la estructura del protocolo SOAP y los elementos con los que está compuesto.

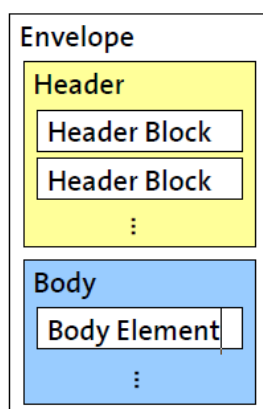


Figura 3: Estructura SOAP
Fuente: (Gustavo, 2008)

1.3.3 UDDI.

Termino que nace de los términos en ingles Universal Description, Discovery and Integration, es una capa que trabaja sobre el protocolo SOAP, tomando en cuenta esto se puede decir que los mensajes de petición y respuesta provistos por SOAP son objetos UDDI.

UDDI posee cuatro niveles de información:

- **Business Entity:**

Corresponde al elemento de nivel superior que provee datos generales de una empresa como dirección, descripción, información de contacto, etc.

- **Business Services:**

Están asociadas a las Business Entity, contiene una descripción del servicio y una lista de categorías como compras, envíos, etc.

- **Binding Template:**

Provee la información técnica necesaria para enlazar e interactuar con los web services. Puede contener entre el punto de acceso para un servicio dado o un mecanismo indirecto que acordara un punto de acceso.

- **tModels:**

Son usados para representar conceptos únicos, Proveen una estructura que permiten la reutilización y de esta forma la estandarización dentro de un framework (Fensel, Bussler, Ding, & Omelayenko, 2005).

1.4 REST (Representational State Transfer)

Es una técnica de arquitectura de software para sistemas y distribuidos como World Wide Web. En REST los datos y las funcionalidades son reconocidos como recursos, estos pueden ser accedidos mediante un identificador único o también llamadas (URI).

REST Utiliza una arquitectura Cliente / Servidor y está diseñado para utilizar un protocolo de comunicación sin estado (stateless), es decir el estado de la conexión se deja de lado, esto con el fin de centrarse en la funcionalidad de los componentes, esto se puede realizar debido a que los mensajes HTTP contienen toda la información que se requiere al entender la petición.

REST destaca el uso de la arquitectura Web, de entre los cuales se enfatiza en los más importantes.

1.4.1 Identificación de Recursos.

Los recursos individuales son identificados en la solicitud (a través de URIs) y son conceptualmente separados del mensaje de respuesta que puede ser HTML, XML, JSON, etc.

1.4.2 Manipulación de Recursos.

Un servicio web cuenta con la posibilidad de manipular los datos siempre y cuando se le provee de un identificador único y una operación a realizar es decir tiene la capacidad de Consultar, Crear, Eliminar y Actualizar ciertos recursos.

1.4.3 Mensajes Autodescriptivos.

Los mensajes que se intercambia entre el cliente y el servidor pueden ser personalizado a través de MIME type, es decir se especifica en el MIME Type el tipo de respuesta y que se desea obtener.

1.4.4 Hipermedia como el motor de las aplicaciones.

Los clientes pueden hacer una transición de un estado a otro a través de acciones que son dinámicamente identificados dentro de la hipermedia por el servidor.

En la Tabla 2. Métodos HTTP utilizados en REST se listan los métodos del protocolo HTTP que son definidos para las interacciones de RESTful.

Tabla 2. Métodos HTTP utilizados en REST

Acción	Descripción
GET:	Es usado por el servicio que realiza una petición para obtener una representación de un recurso desde una URI.
DELETE:	Usado para eliminar representaciones de un recurso.
POST:	Actualiza o crea la representación de un recurso.
PUT:	Crear representaciones de un recurso (Saquicela, Vilches-Blázquez, & Corcho, 2012).

Fuente: (Saquicela, Vilches-Blázquez, & Corcho, 2012)

1.5 Jersey

Jersey es un proyecto que usa un framework para la construcción de Servicios Web RESTful utilizando tecnología Java, es un proyecto de código abierto y proporciona soporte para los APIs de JAX-RS y JAX-RS (JSR 311 y JSR 339).

Componentes.

- Core Server
- Core Client
- JAXB Support
- JSON Support
- Módulo de integración para Spring y Guice

1.6 Jackson

Jackson es una librería multiproceso de Java que permite el procesamiento del formato JSON, es decir que ayuda a la creación, lectura y serialización de datos Json en aplicaciones Java. Jersey usa la librería Jackson para gestionar y responder respuestas tipo JSON.

1.7 JSON

JSON (JavaScript Object Notation) es un formato de datos liviano para el intercambio de datos. Es sencillo de entender para las personas al leer y escribir. Su basa en un subconjunto del lenguaje de programación JavaScript, estándar ECMA-262 3rd Edición - Diciembre 1999. JSON es un formato de texto que es totalmente independiente del lenguaje, pero utiliza las convenciones que son familiares para los programadores de C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros más. Estas propiedades hacen JSON un ideal de intercambio de datos de idiomas.

1.7.1 Estructura JSON.

- Una colección de pares clave / valor. En varios lenguajes, esto se realiza como un objeto, registro, estructura, diccionario, tabla hash, lista con llave, o una matriz asociativa.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se realiza como una array, vector, lista o secuencia.

Estas son estructuras de datos universales. Prácticamente todos los lenguajes de programación modernos lo usan de una forma u otra. Tiene sentido que un formato de datos que es intercambiable con los lenguajes de programación también se basan en estas estructuras (JSON, 2010).

1.8 Linked Data

Creada con el objetivo de que la información que está disponible en internet sea de mayor provecho para las personas interesadas de esta tecnología y por las máquinas y buscadores que ayudan a filtrar la gran cantidad de contenidos que se encuentran en la red.

Internet se ha convertido en la fuente de consulta y de referencia para la mayoría de las personas, actualmente en la red se puede encontrar mucha información confusa y desordenada esto ocasiona que la indexación de buscadores y robot produzcan pocos resultados y sin sentido, por este motivo al organizar la información de una forma correcta se garantiza por ejemplo en una búsqueda la exclusión de contenidos poco importantes, va ser más rápida y precisa para los robots que hacen este tipo de filtros debido a que cuentan con la suficiente información para decidir cuál resultado está mejor relacionado con lo buscado.

Linked Data describe las mejores prácticas para publicar y conectar datos estructurados en la web (Bizer, Heath, & Berners-Lee, 2009), en resumen Linked Data describe un método para publicar datos estructurados, ser entrelazados y organizar la información y de esta forma ser de mucha utilidad para los usuarios, este método está construido sobre los

estándares de las tecnologías web HTTP3 y URIs4 regulados por la W3C5 (World Wide Web Consortium), de esta forma se puede mostrar la información como páginas web para que las personas puedan leer y entender, pero también implica compartir la información de manera que pueda ser leída de forma automática por los computadores.

1.9 Big Data

Big Data es el término que se le ha denominado a grandes bases de Datos y DataSets esto debido al volumen de los datos que contienen, dificultando tareas como con las búsquedas, compartición, análisis, procesamiento y visualización.

Sin embargo Big Data no solo hace referencia a grandes cantidades de datos, hoy en día para que los datos sean considerados como Big Data es necesario que cumpla con un conjunto de reglas que se denominan las 4 Vs del Big Data, estas reglas fueron propuestas por IBM en el 2012 y describe cuatro condiciones que tienen que cumplir los datos para que sean considerados como Big Data.

1.9.1 Volumen.

Es la una de las características más importantes del Big Data, cuando se habla de volumen se habla de la cantidad de la información y el tamaño que ocupa los datos que están por encima de los terabytes y petabytes, tamaños que no se puede almacenar en dispositivos de almacenamiento tradicional. (Sagiroglu & Sinanc, 2013)

Las necesidades que surgen cuando se trabaja con grandes volúmenes de datos son que estos estén accesibles, que se pueda buscar algún dato en específico entre todo el volumen de datos y además proveer la posibilidad de realizar operaciones y procesamiento sobre ellos.

1.9.2 Variedad.

La variedad de información hace que el volumen de los datos sea realmente grande. Big Data se conforma por una variedad de información que puede venir de diferentes fuentes y recursos.

En cuanto a la variedad de los datos se puede tener de tres tipos:

1.9.2.1 Estructurados.

Los datos estructurados son aquellos en los cuales ya se ha realizado un análisis previo y que por lo general ya se encuentran en datawarehouse, estos datos han sido clasificados, etiquetados y almacenados en algún dispositivo o servicio (Sagiroglu & Sinanc, 2013)

1.9.2.2 No Estructurados.

Los datos no estructurados son aquellos que no se ha realizado ninguna operación sobre ellos y por lo general están desordenados y mezclados entre sí, por lo cual es complicado realizar cualquier análisis sobre ellos.

1.9.2.3 Semiestructurados.

Los datos semiestructurados son aquellos que de alguna forma ya cuentan con una estructura por los cuales se han clasificados pero aún no existe la definición exacta de los campos, rango y tipo de datos.

1.9.3 Velocidad.

Debido a la variedad de la información el volumen de los datos puede aumentar considerablemente en poco tiempo, la velocidad es una característica importante no solo cuando se trabaja con Big Data si no para cualquier proceso y sobre todo cuando se trabaja con procesos de tiempo limitado lo cual requiere el procesamiento de los datos en tiempo real o de lo contrario la solicitudes de datos se pierden. Trabajar con grandes datos con tiempos de respuesta aceptables maximiza la productividad y el valor de cualquier organización. (Sagiroglu & Sinanc, 2013)

1.9.4 Veracidad.

Veracidad es un aspecto importante de los datos ya que depende de esta característica que den resultados de confianza a los usuarios. La veracidad asegura que los datos con los que se están trabajando son reales auténticos y protegidos de la manipulación de personas no autorizadas. Los datos deben de ser seguros en todo el ciclo de vida, desde la extracción de fuentes de confianza así como el procesamiento en instituciones y usando herramientas seguras. (Demchenko, Paola, de Laat, & Membrey, 2013)

Cuando se habla de veracidad refiriéndose al Big Data se tiene dos aspectos a considerar:

Consistencia o certeza de los datos, significa que los datos y sus valores no han sido alterados y que se ha seguido un modelo óptimo para evitar que esto suceda.

Integridad de los datos, hace referencia que en el proceso de extracción los datos no han sido alterados en algún proceso, es definido por el número de factores que involucra el origen de los datos, colecciones de datos y métodos de procesamiento, incluyendo el uso de una infraestructura de confianza y la factibilidad en la que se realizó.

Para asegurar la veracidad de los datos son necesarios los siguientes aspectos:

- Integridad de los datos y datos enlazados

- Autenticidad de los datos y orígenes de datos de confianza
- Identificación de los datos y su origen.
- Procesamiento y almacenamiento en plataformas de confianza.
- Disponibilidad de los datos
- Responsabilidad y buena reputación.

1.10 Virtuoso

Virtuoso es un servidor de datos multi-modelo empresarial. Ofrece una solución de alta calidad, es una plataforma incomparable para la gestión de datos, el acceso y la integración.

Virtuoso es una base de datos de alto rendimiento con una compatibilidad con muchos de los lenguajes de programación actualmente existentes, permite hacer consultas mucho más inteligentes y veloces que otros productos del mercado.

Virtuoso provee dos formas de importar datos al formato RDF, la primera consiste en estructurar los datos de tal forma que se obtengan tripletas y hacer una inserción directa al triple store; la segunda opción consiste en hacer una conversión de las consultas SPARQL a SQL este proceso también se lo conoce como *reescritura de SQL*, el objetivo consiste en preparar el resultado en formato RDF (Erling).

Cuando se trabaja con tripletas por lo general se habla de un RDF graph que es una representación visual de las tripletas que conforman el RDF. Una tripleta es una forma de representar una sentencia usada por RDF que consta de una propiedad, el identificador del recurso, y el valor de la propiedad (Sujeto, Predicado y Objeto); frente a esto es necesario olvidar por un momento de las bases de datos relacionales debido a que principalmente no se tiene tablas ni relaciones, se trabaja sobre una única instancia que se denomina Graph.

1.11 SPARQL

Es un lenguaje de consulta similar a SQL de Base de Datos para poder hacer búsquedas de información en la nube de enlaces, este puede ser usado para expresar consultas a diferentes fuentes de datos, si los datos están almacenados nativamente en RDF o vistos como RDF mediante un middleware. La respuesta a las consultas lanzadas con este lenguaje suelen dar como resultado un JSON o un XML como muchos otros servicios Web

1.12 OER

La definición de los OER en la actualidad se utiliza con más frecuencia es "material digital ofrecidos libre y abiertamente a los docentes educadores, estudiantes y autodidactas para

utilizar y reutilizar, para la enseñanza, el aprendizaje y la investigación". OER incluye contenidos de aprendizaje, herramientas de software para desarrollar, usar y distribuir contenido, y los recursos de implementación, tales como las licencias libres (CENTRE FOR EDUCATIONAL RESEARCH AND INNOVATION, 2007).

El término recurso educativo abierto es el resultado de una reunión de la primavera de 2002 se celebró en la UNESCO, y organizado con el apoyo de WCET y la Fundación William y Flora Hewlett.

1.13 Visualización de Datos

“Los datos, organizados y empleados debidamente, pueden convertirse en información.

La información, absorbida, comprendida y aplicada por las personas, puede convertirse en conocimientos.

Los conocimientos aplicados frecuentemente en un campo pueden convertirse en sabiduría, y la sabiduría es la base de la acción positiva.” (Cooley, 1987)

Hoy en día el conocimiento es un elemento muy importante para la sociedad y ha venido aumentando masivamente en los últimos tiempos por lo que cada vez se necesita mucho más tiempo para transmitir el nuevo y complejo conocimiento, es por esta razón que es necesario buscar alternativas que permitan transmitir este conocimiento de manera rápida y sencilla. (Meyer, 2009)

Visualización de Datos es la representación gráfica de la información abstracta para el análisis de los datos y la comunicación (Few, 2013).

El propósito de la Visualización de Datos es poder adquirir conocimiento a partir del análisis de los datos, para esto es necesario pasar por un proceso de complejidad cognitiva, que constan de las siguientes fases:

- **Datos:** Representan un conjunto de símbolos, caracteres alfanuméricos que no tienen relación ni significado por si solos.
- **Información:** Son un conjunto de datos elaborados, este tipo de dato ya han sido procesados e interpretados y normalmente pueden dar respuesta a preguntas como: “Quien”, “Que”, “Cuando”, “Donde”, “Por qué”. (Keller & Tergan, 2005)
- **Conocimiento:** El conocimiento esta un paso delante de la Información pues ha sido cognitivamente procesado es decir que los conceptos detrás de la información han sido asimilados, Una forma de distinguir la Información del Conocimiento es que la

información es algo que se encuentra fuera nuestro y el conocimiento es algo que se encuentra dentro como un concepto entendido claramente. (Keller & Tergan, 2005)

- **Sabiduría:** Se refiere a tener un completo entendimiento del conocimiento adquirido, a poder reflexionar y sacar conclusiones en base a un buen sentido de lo que es correcto, es un elemento intransferible ya que nace con la aplicación del conocimiento y la experiencia propia. (Wikipedia, 2010)

La visualización de datos es una rama de la investigación que nace con la intención de transformar datos en información haciendo uso de diferentes herramientas de visualización.

1.13.1 ¿Por qué se visualiza?

Los datos son importantes ya que detrás de ellos se esconden un sin número de información que puede explicar muchas cosas o contar una historia (Few, 2013), siempre y cuando estos estén correctamente estructurados de forma delicada y ordenada, por ejemplo si se cuenta con un conjunto de datos numéricos estos pueden representar muchas cosas como notas académicas, montos de alguna compra, pesos, tallas incluso coordenadas geográficas. La visualización de datos es una herramienta significativa ya que ayuda en el proceso de descubrir y entender la aplicación de estos y las historias que están por detrás.

La Visualización de Datos es una forma de representar algo abstracto en algo más sencillo y fácil de comprender por lo cual se ha convertido en una nuevo campo de la investigación, cuyo propósito es el de buscar nuevas formas de transmitir el conocimiento (Meyer, 2009).

En la actualidad existen muchas formas de representar los datos visualmente y se puede utilizar cualquier medio para ello, sin embargo no todas son apropiadas para generar nuevas ideas y conocimiento, necesarios para la comprensión y la transferencia de conocimiento (Meyer, 2009).

La visualización de datos ayuda en el proceso de aprendizaje ya que el proceso de percepción humana es más rápido visualizando que el proceso cognitivo que hace el cerebro para interpretar los datos; el análisis tradicional de los datos requiere un análisis consiente para la mayoría de trabajos, la visualización de datos cambia esto ya que hace mayor uso de la percepción visual aprovechando todo lo que se puede ver (Few, 2013).

En 1972 un estudio realizado por Gestalt School of Psychology busco la forma de explicar cómo el cerebro humano buscaba patrones, formas y organiza lo que se ve, el resultado fue un conjunto principios de percepción que todavía se aplican en este tiempo; En la Figura 4: Patrones de percepción humana se detalla algunos de ellos.

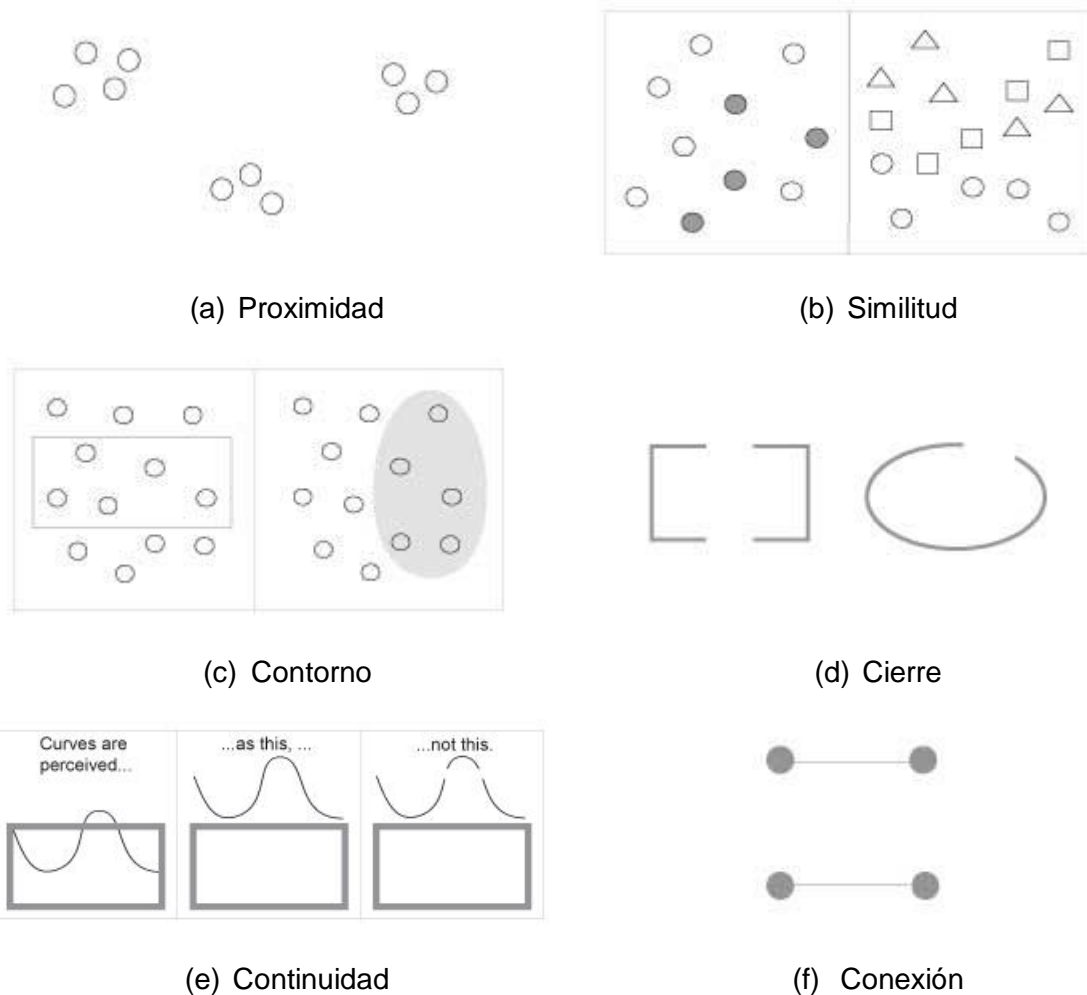


Figura 4: Patrones de percepción humana
 Fuente: (Few, 2013)
 Elaboración: Jonathan Yaguachi, Mayo 2015

El cerebro humano de forma directa o indirecta está preparado para reconocer patrones en lo que se visualiza asociadas a experiencias pasadas, es posible hacer usos de estos patrones para transmitir un mensaje de forma directa y rápida reduciendo tiempo que normalmente se usa para interpretar y asimilar los datos por si solos.

1.14 Patrón de Visualización

Dentro de un conjunto de datos un patrón de visualización representa los elementos y formas visuales que usan en una gráfica para representar una estructura de datos en específico, la estructura básica de información la cual actúa como plantilla o template, a partir de este se puede estructurar un conjunto de datos en un solo formato para el intercambio de mensajes e integración de módulos. En este caso se utiliza patrones con notación JSON.

En la web existen algunas librerías que se usan para realizar Visualización de Datos, se buscaron y analizaron algunas de ellas y se describen a continuación.

1.15 Exhibit¹

Exhibit es una librería creada por MIT y es un proyecto de código abierto que permite la creación de mapas interactivos y otro tipo de visualizaciones orientadas hacia la enseñanza de un conjunto de datos estáticos y geográficos como la ubicación de un pin que identifique cada país o mostrar ciertos lugares de interés, así mismo también permite funcionalidades como búsquedas y filtrado de texto.

1.16 JavaScript InfoVis Toolkit²

InfoVis es una librería muy interesante, ya que presentan una estructura modular, es decir no muestra todo el conjunto de datos, muestra el resultado que el usuario desea ver a nivel general, se puede desplegar el grafo hasta alcanzar un nivel más bajo o hasta que el usuario desee, esta librería cuenta con estilos y efectos de animación únicos y es un proyecto de código abierto.

1.17 Arbor.js³

Arbor es una librería de visualización que fue creada para trabajar en la web y que está basada en la librería de JavaScript jQuery. Arbor no posee múltiples diseños y plantillas para visualizar si no que más bien se enfoca únicamente en el algoritmo de diseño “Force-directed⁴”, la organización gráfica y la correcta presentación en pantalla de los resultados.

1.18 D3.js⁵

Es una librería de JavaScript que trabaja en conjunto con HTML, SVG y CSS para presentar graficas con un buen diseño y estilo, se puede utilizar una variedad de formatos de datos como CVS, DSV, TSV, JSON, etc. Es capaz de presentar visualizaciones avanzadas con un conjunto de datos complejos y es un proyecto de código abierto por lo cual tiene una gran comunidad de usuarios que apoya con el desarrollo, soporte y documentación del proyecto.

1.19 SigmaJS⁶

Es una librería basada en JavaScript que permite graficar en el navegador diferentes grafos, esta librería se caracteriza por trabajar con archivos con extensión .gexf, el mismo formato

¹ Exhibit (<http://www.simile-widgets.org/exhibit/>)

² JavaScript InfoVis Toolkit (<http://philogb.github.io/jit/>)

³ Arbor.js (<http://arborjs.org/>)

⁴ https://en.wikipedia.org/wiki/Force-directed_graph_drawing


⁵ D3.js (<http://d3js.org/>)

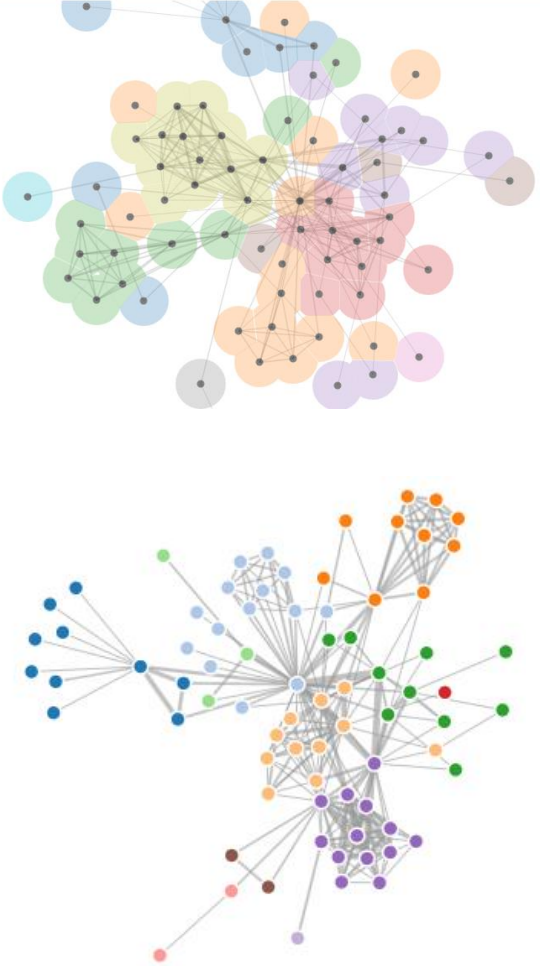
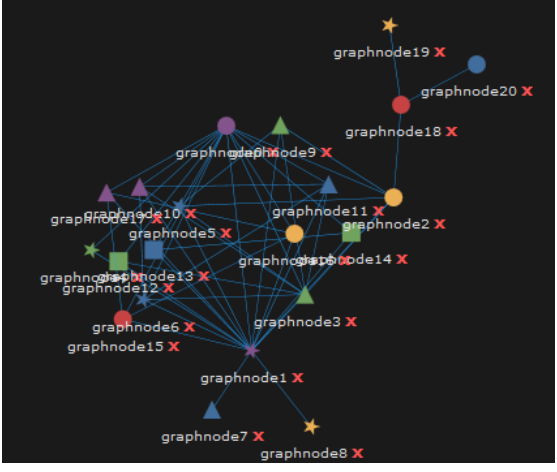
⁶ SigmaJS (<http://sigmajs.org/>)

que exporta el software de visualización Gephi por lo tanto esta librería se suele utilizar para visualizar en la web, trabajos realizados previamente en Gephi.

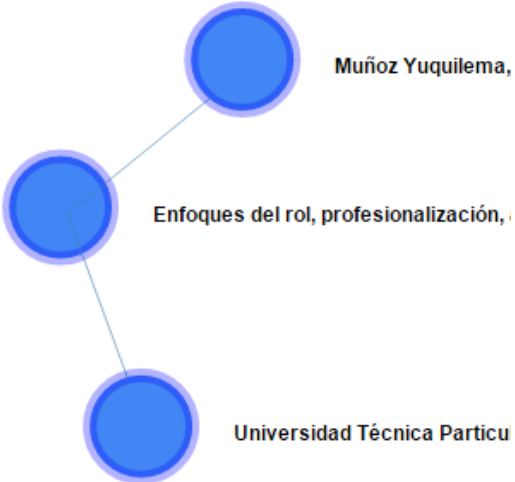
En la Tabla 3. Formato JSON de las herramientas seleccionadas se presenta las gráficas analizadas así como la estructura y formato de JSON con el que trabajan.

Tabla 3. Formato JSON de las herramientas seleccionadas

Librería	Grafica	Patrón
Arbor		<pre> { "nodes": { "WA": { }, "DE": { }, "DC": { } }, "edges": { "WA": { "DE": { }, "DC": { } } } } </pre>

<p style="text-align: center;">D3</p>		<pre> { "nodes": [{ "name": "Myriel", "group": 1 }, { "name": "Napoleon", "group": 2 }, { "name": "Mlle.Baptistine", "group": 2 }], "links": [{ "source": 1, "target": 0, "value": 1 }, { "source": 2, "target": 0, "value": 100 }] } </pre>
<p style="text-align: center;">JavaScript InfoVis Toolkit</p>		<pre> { "adjacencies": [{ "nodeTo": "graphnode1", "nodeFrom": "graphnode0", "data": {} }, { "nodeTo": "graphnode3", "nodeFrom": "graphnode0", "data": {} }], "data": { "\$color": "#83548B", "\$type": "circle" }, "id": "graphnode0", "name": "graphnode0" } </pre>

<p>D3</p>		<pre>{ "name": "flare", "children": [{ "name": "analytics", "children": [{ "name": "AgglomerativeCluster" }, { "name": "CommunityStructure" }] }] }</pre>
<p>D3</p>		<pre>{ "name": "flare", "children": [{ "name": "analytics", "children": [{ "name": "AgglomerativeCluster", "size": 2323 }, { "name": "CommunityStructure", "size": 2324 }] }] }</pre>
<p>JavaScript InfoVis Toolkit</p>		<pre>{ "id": "node37", "name": "3.7", "data": { }, "children": [{ "id": "node48", "name": "4.8", "data": { }, "children": [] }] } }</pre>

D3		<pre> { "name": "flare", "children": [{ "name": "analytics", "children": [{ "name": "AgglomerativeCluster", "size": 2323 }, { "name": "CommunityStructure", "size": 2324 }] }] } </pre>
----	---	---

Fuente: Jonathan Yaguachi

Elaboración: Jonathan Yaguachi, Junio 2015

1.20 Trabajos relacionados

1.20.1 La visualización de resultados de búsqueda y los gestores bibliográficos personales: una nota técnica sobre RefViz 2

Con la llegada de las Bases de Datos científicas y su distribución a través de la Web con el tiempo ha supuesto un reto así como también una oportunidad el manejo de información científica.

Tanto los grupos primarios (editores), como los secundarios (las bases de datos de bibliografía científica) han evolucionado y se han adapta al creciente número de publicaciones producidos en la actualidad que cada día crece más y más.

La visión actual de la información ha provisto de un nuevo enfoque que está inspirada en la web, lo ha traído nuevas y mejores oportunidades sobre el manejo de la información científica que va más allá de la de la simple ordenación lineal de múltiples documentos que son el resultado de búsquedas.

Tomando en cuenta este nuevo contexto los documentos no se presentan de forma aislada sino más bien interrelacionada entre sí, y aun mejor estos elementos de relación se pueden utilizar en el filtrado y ordenación de los grupos de documentos buscados.

Los principios básicos para la organización de documentos científicos son los siguientes:

- Todo documento está relacionado entre si y esta relación depende de su estructura.
- La cuantificación de estas relaciones permiten ordenar los documentos en espacios informativos. (Amat, 2009)

Las bases de datos científicas relacionan sus documentos publicados convirtiendo sus relaciones a vínculos que apuntan a otros documentos con la misma temática y algunas de las veces con algunas expresiones graficas lo que proporcionan un sentido de contexto en el recorrido de grandes bases de datos de bibliografía científica.

RefViz

Su funcionamiento empieza con el mecanismo de carga y visualización de referencias. El objetivo es la estructuración de grupos de documentos científicos a través de análisis de sus textos, habitualmente sus resúmenes.

Permite visualizar los documentos agrupados, su lista y términos principales y secundarios resultantes por el análisis realizado por el programa. Se puede escoger entre dos formatos de visualización de los conjuntos de referencias:

- Representación tabular (Matriz)
- Representación espacial (Galaxia)

El proceso siguiente que se debe realizar es el descarte de alguno de los términos principales o el impulso de los términos secundarios con el objetivo de reordenar los grupos de documentos y acelerar la identificación de referencias.

La visualización de los datos que se obtiene ayuda al usuario a sacar una conclusión a simple vista sobre el análisis de un determinado tema.

CAPÍTULO II: DESCRIPCIÓN DE DATOS

2.1 Introducción

En el presente capítulo se describe de forma detallada los datos con los cuales se trabajaran; origen de los datos y dirección en donde se encuentran disponibles, repositorio y tecnologías que se usaron así como también el formato de los mismos. De la misma manera se presenta la descripción básica de un grafo RDF y sus elementos, elementos como las tripletas sujeto, predicados y objeto y su definición. Finalmente se detalla el concepto de vocabulario semántico y su utilidad en el grafo RDF, describiendo los vocabularios utilizados en el EndPoint y para mayor entendimiento estos se los representa gráficamente para representar la relación entre el sujeto predicado y objeto.

2.2 Datos a trabajar

Para la creación de la aplicación de visualización de datos se usó un EndPoint de Virtuoso creado por la UTPL y en donde se almacenan la información de trabajos de fin de titulación de estudiantes de la universidad tanto de pregrado como postgrado y en mínimas cantidades otros recursos como libros, videos, imágenes, etc.

El EndPoint se puede acceder en la dirección <http://data.utpl.edu.ec/serendipity/oar/sparql> y se trabajó exclusivamente con los datos que se encuentra dentro del Grafo <http://data.utpl.edu.ec/serendipity/oar>. Las consultas al EndPoint se realizan mediante SPARQL y dispone de una variedad de formatos a devolver la información consultada como HTML, XML, JSON, NTriples, RDF/ML, CSV, TSV, etc. Para la construcción de la aplicación se usó el formato de datos JSON.

Los datos provistos trabajan bajo el protocolo OAI⁷ (Open Archives Initiative) una iniciativa creada con el objetivo de promover estándares de interoperabilidad para proveer la transmisión eficiente de contenidos en internet, no solamente de publicaciones científicas si no de cualquier recurso almacenado en digital (Barrueco & Subirats Coll, 2003).

La OAI no especifica o define ninguna regla para la gestión de los derechos; siendo el proveedor de los datos el responsable del acceso y la gestión de la propiedad intelectual de los datos publicados.

Esta información ya ha sido previamente elaborada por lo cual ya se cuenta con un rdf graph, que implica que los datos se encuentran en tripletas Sujeto, Predicado y Objetos y que cada tripleta tiene asignado un predicado que relacione el sujeto con el objeto.

⁷ <https://www.openarchives.org/>

2.2.1 Sujeto.

Dentro del RDF los sujetos se los registra e idéntica como nodos, pueden ser una referencia URIs o simplemente pueden ser nodos vacíos (W3C, 2004).

2.2.2 Predicado.

Los predicados pueden ser interpretados como una relación entre dos nodos o como la definición de un valor o atributo (Objeto) para un determinado Sujeto, también se los conoce como sinónimos o propiedad de una tripleta; su valor es una referencia URI (W3C, 2004).

2.2.3 Objeto.

Al igual que el Sujeto, el Objeto puede ser identificado como un nodo y los valores que pueden contener son una referencia de una URI, un literal o un nodo vacío (W3C, 2004).

Una representación gráfica de una tripleta con sus tres elementos se muestra en la Figura 5. Representación Gráfica de una Tripleta

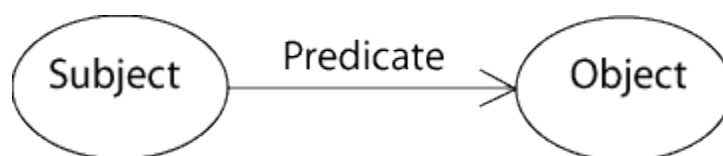


Figura 5. Representación Gráfica de una Tripleta
Fuente: (W3C, 2004)

2.3 Vocabularios

Cuando se trabaja con tripletas, es decir con elementos sujeto, predicado y objeto se ve la necesidad de la utilización de vocabularios y de ser necesario, la definición de uno nuevo.

Los vocabularios ayudan a relacionar dos nodos bajo un concepto por ejemplo si se quiere relacionar un Autor con un trabajo de fin de titulación la tripleta sería como se muestra en la Tabla 4. Ejemplo Tripleta

Tabla 4. Ejemplo Tripleta

Sujeto	Predicado	Objeto
http://data.utpl.edu.ec/sere/ndipity/oar/resource/0004b8e38b2dbfcb042704fe2c2b5ee9	http://purl.org/dc/terms/creator	http://data.utpl.edu.ec/sere/ndipity/oar/resource/Abril Fausto, Idrovo

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Enero 2015

El predicado define una propiedad o características que contiene el sujeto y cuyo valor está representado en el objeto.

<http://data.utpl.edu.ec/serendipity/oar/resource/0004b8e38b2dbfcb042704fe2c2b5ee9> tiene un <http://purl.org/dc/terms/creator> que es http://data.utpl.edu.ec/serendipity/oar/resource/Abril_Fausto_Idrovo.

En la Figura 6. Tripleta, relación Autor – Publicación. Para el ejemplo se usó el elemento **creator** de Dublin Core. Dublin Core es un conjunto de Elementos que se usan para describir semánticamente cualquier recurso en forma de metadatos usando 15 definiciones semánticas.

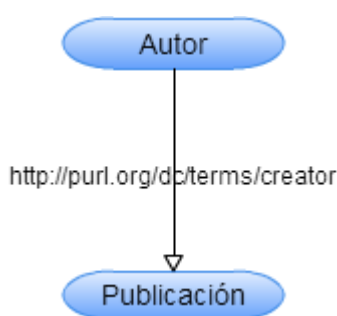


Figura 6. Tripleta, relación Autor – Publicación
Fuente: Datos Serendipity
Elaboración: Jonathan Yaguachi

El objetivo de los Predicados es la reutilización de recurso compartidos por otros usuarios en trabajos anteriores, cualquier persona puede descubrir y reutilizar vocabularios compartidos en la Web y adaptarlos de acuerdo al trabajo que se esté realizando (Lamarca Lapuente, 2013).

Buscar un vocabulario cuyo concepto describa de forma correcta con cada uno de los valores de los datos obtenidos es una tarea fundamental para armar las tripleteas, a continuación se describe los vocabularios con los que están trabajando los datos.

2.3.1 RDF.

Vocabulario RDF es un conjunto de IRIs que está destinado para el uso de rdfs graphs, puede ser usado para definir y documentar un vocabulario RDF adicional. Los términos disponibles se encuentran descritos en la Tabla 5. Términos RDF.

Tabla 5. Términos RDF

Type	Namespace	Term
property	http://purl.org/dc/elements/1.1/	title

property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	type
class	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Property
class	http://www.w3.org/2000/01/rdf-schema#	Class
property	http://www.w3.org/2000/01/rdf-schema#	isDefinedBy
property	http://www.w3.org/2000/01/rdf-schema#	label
property	http://www.w3.org/2000/01/rdf-schema#	comment
property	http://www.w3.org/2000/01/rdf-schema#	subClassOf
property	http://www.w3.org/2000/01/rdf-schema#	range
property	http://www.w3.org/2000/01/rdf-schema#	domain
property	http://www.w3.org/2000/01/rdf-schema#	subPropertyOf
property	http://www.w3.org/2000/01/rdf-schema#	seeAlso
class	http://www.w3.org/2002/07/owl#	Ontology

Fuente: (Graphite; ARC2)

2.3.2 Rdf-schema.

Rdf schema (rdfs) es una extensión básica del vocabulario RDF. Provee un vocabulario de modelado de datos para RDF. En la Tabla 6. Términos RDFS describe los términos de los cuales se pueden hacer usos para describir datos.

Tabla 6. Términos RDFS

Tipo	Namespace	Termino
property	http://purl.org/dc/elements/1.1/	title
property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	type
class	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Property
class	http://www.w3.org/2000/01/rdf-schema#	Class
property	http://www.w3.org/2000/01/rdf-schema#	isDefinedBy
property	http://www.w3.org/2000/01/rdf-schema#	label
property	http://www.w3.org/2000/01/rdf-schema#	comment
property	http://www.w3.org/2000/01/rdf-schema#	subClassOf
property	http://www.w3.org/2000/01/rdf-schema#	range
property	http://www.w3.org/2000/01/rdf-schema#	domain
property	http://www.w3.org/2000/01/rdf-schema#	subPropertyOf
property	http://www.w3.org/2000/01/rdf-schema#	seeAlso
class	http://www.w3.org/2002/07/owl#	Ontology

Fuente: (Graphite; ARC2)

2.3.3 SKOS.

Simple Knowledge Organization System (SKOS) es un modelo de datos común para enlazar y compartir sistemas de organización de conocimiento a través de la web semántica, los elementos que conforman el vocabulario SKOS se describen en la Tabla 7. Términos SKOS

Tabla 7. Términos SKOS

Type	Namespace	Term
property	http://purl.org/dc/terms/	title
property	http://purl.org/dc/terms/	contributor
property	http://purl.org/dc/terms/	description
property	http://purl.org/dc/terms/	creator
property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	type
class	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Property
property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	first
property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	rest
property	http://www.w3.org/2000/01/rdf-schema#	seeAlso
property	http://www.w3.org/2000/01/rdf-schema#	label
property	http://www.w3.org/2000/01/rdf-schema#	isDefinedBy
property	http://www.w3.org/2000/01/rdf-schema#	subClassOf
property	http://www.w3.org/2000/01/rdf-schema#	range
property	http://www.w3.org/2000/01/rdf-schema#	domain
property	http://www.w3.org/2000/01/rdf-schema#	subPropertyOf
property	http://www.w3.org/2000/01/rdf-schema#	comment
class	http://www.w3.org/2002/07/owl#	Ontology
class	http://www.w3.org/2002/07/owl#	Class
property	http://www.w3.org/2002/07/owl#	disjointWith
class	http://www.w3.org/2002/07/owl#	ObjectProperty
property	http://www.w3.org/2002/07/owl#	inverseOf
class	http://www.w3.org/2002/07/owl#	AnnotationProperty
class	http://www.w3.org/2002/07/owl#	DatatypeProperty
class	http://www.w3.org/2002/07/owl#	SymmetricProperty
class	http://www.w3.org/2002/07/owl#	TransitiveProperty
property	http://www.w3.org/2002/07/owl#	unionOf
class	http://www.w3.org/2002/07/owl#	FunctionalProperty
property	http://www.w3.org/2004/02/skos/core#	definition
property	http://www.w3.org/2004/02/skos/core#	scopeNote

property	http://www.w3.org/2004/02/skos/core#	example
----------	--------------------------------------	---------

Fuente: (Graphite; ARC2)

2.3.4 Dublin Core

Es un conjunto definiciones semánticas o metadatos que facilitan la tarea de descubrir recursos electrónicos (Weibel, 1997); originalmente fue creada para describir autores de recursos web pero ahora también se puede describir organizaciones educativas, bibliotecas, instituciones gubernamentales, museos, librerías, etc. (Wikipedia, 2013).

Normalmente se los suele clasificar en tres grupos:

- Por su contenido
- Por la Propiedad Intelectual
- Instanciación

Tabla 8. Términos Dublin Core

Type	Namespace	Term
class	http://purl.org/dc/dcam/	VocabularyEncodingScheme
property	http://purl.org/dc/terms/	modified
property	http://purl.org/dc/terms/	publisher
property	http://purl.org/dc/terms/	title
property	http://purl.org/dc/terms/	description
property	http://purl.org/dc/terms/	hasVersion
property	http://purl.org/dc/terms/	issued
class	http://purl.org/dc/terms/	AgentClass
property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	type
class	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Property
class	http://www.w3.org/2000/01/rdf-schema#	Class
property	http://www.w3.org/2000/01/rdf-schema#	comment
property	http://www.w3.org/2000/01/rdf-schema#	isDefinedBy
property	http://www.w3.org/2000/01/rdf-schema#	label
property	http://www.w3.org/2000/01/rdf-schema#	subClassOf
class	http://www.w3.org/2000/01/rdf-schema#	Datatype
property	http://www.w3.org/2000/01/rdf-schema#	seeAlso
property	http://www.w3.org/2000/01/rdf-schema#	subPropertyOf
property	http://www.w3.org/2000/01/rdf-schema#	range
property	http://www.w3.org/2000/01/rdf-schema#	domain

property	http://www.w3.org/2002/07/owl#	equivalentProperty
property	http://www.w3.org/2004/02/skos/core#	note

Fuente: (Graphite; ARC2)

2.3.5 FOAF.

Vocabulario Friend of a Friend (FOAF), es un proyecto para enlazar personas y su información en la web independientemente si esta información se encuentra de forma física en documentos o si se encuentra en bases de datos, cuenta con algunas propiedades las cuales se describen en la Tabla 9. Términos FOAF.

Tabla 9. Términos FOAF

Type	Namespace	Term
property	http://purl.org/dc/elements/1.1/	title
property	http://purl.org/dc/elements/1.1/	description
property	http://www.w3.org/1999/02/22-rdf-syntax-ns#	type
class	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Property
class	http://www.w3.org/2000/01/rdf-schema#	Class
property	http://www.w3.org/2000/01/rdf-schema#	label
property	http://www.w3.org/2000/01/rdf-schema#	comment
property	http://www.w3.org/2000/01/rdf-schema#	isDefinedBy
property	http://www.w3.org/2000/01/rdf-schema#	subClassOf
property	http://www.w3.org/2000/01/rdf-schema#	domain
property	http://www.w3.org/2000/01/rdf-schema#	range
property	http://www.w3.org/2000/01/rdf-schema#	subPropertyOf
class	http://www.w3.org/2002/07/owl#	Ontology
class	http://www.w3.org/2002/07/owl#	AnnotationProperty
class	http://www.w3.org/2002/07/owl#	Class
property	http://www.w3.org/2002/07/owl#	equivalentClass
property	http://www.w3.org/2002/07/owl#	disjointWith
class	http://www.w3.org/2002/07/owl#	InverseFunctionalProperty
class	http://www.w3.org/2002/07/owl#	ObjectProperty
class	http://www.w3.org/2002/07/owl#	DatatypeProperty
class	http://www.w3.org/2002/07/owl#	FunctionalProperty
property	http://www.w3.org/2002/07/owl#	inverseOf
property	http://www.w3.org/2002/07/owl#	equivalentProperty
property	http://www.w3.org/2003/06/sw-vocab-status/ns#	term_status

Fuente: (Graphite; ARC2)

2.4 Clasificación de los Datos

Como se ha mencionado anteriormente los datos que se encuentran almacenados en el EndPoint son recursos abiertos publicados por la Universidad Técnica Particular de Loja los mismos que en virtuoso se los puede clasificar como lo muestra la Tabla 10. Tipos de Datos Virtuoso.

Tabla 10. Tipos de Datos Virtuoso

Tipo	Concepto
Persona	http://xmlns.com/foaf/0.1/Person
Concepto	http://www.w3.org/2004/02/skos/core#Concept
Tesis	http://purl.org/ontology/bibo/Thesis
	http://purl.org/ontology/bibo/ThesisDegree
	http://purl.org/dc/terms/BibliographicResource
Maestría	http://purl.org/net/nknouf/ns/bibtex#Mastersthesis
	http://purl.org/ontology/bibo/Thesis
	http://purl.org/dc/terms/BibliographicResource
Video	http://purl.org/ontology/bibo/AudioVisualDocument
	http://schema.org/VideoObject
	http://vivoweb.org/ontology/core#Video
	http://purl.org/dc/terms/BibliographicResource
Libro	http://purl.org/ontology/bibo/Book
	http://schema.org/Book
	http://purl.org/dc/terms/BibliographicResource
Reporte	http://purl.org/net/nknouf/ns/bibtex#Techreport
	http://purl.org/ontology/bibo/Report
	http://purl.org/dc/terms/BibliographicResource
Documento	http://purl.org/ontology/bibo/Document
	http://purl.org/dc/terms/BibliographicResource
Presentación	http://purl.org/ontology/bibo/Slideshow
	http://vivoweb.org/ontology/core#Presentation
	http://purl.org/dc/terms/BibliographicResource
Artículo	http://schema.org/Article
	http://purl.org/ontology/bibo/AcademicArticle
	http://purl.org/dc/terms/BibliographicResource
Recurso de Aprendizaje	http://schema.org/learningResourceType
	http://purl.org/dc/terms/BibliographicResource

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

Según los valores presentados en la Tabla 10. Tipos de Datos Virtuoso, se define los tipos de recursos disponibles en el EndPoint y de cuales se usaran en la aplicación de visualización.

2.5 Representación Grafica de los Vocabularios

A continuación se muestran de forma gráfica los vocabularios usados para cada uno de los tipos de recursos disponibles en el EndPoint de Virtuoso Serendipity OAR⁸.

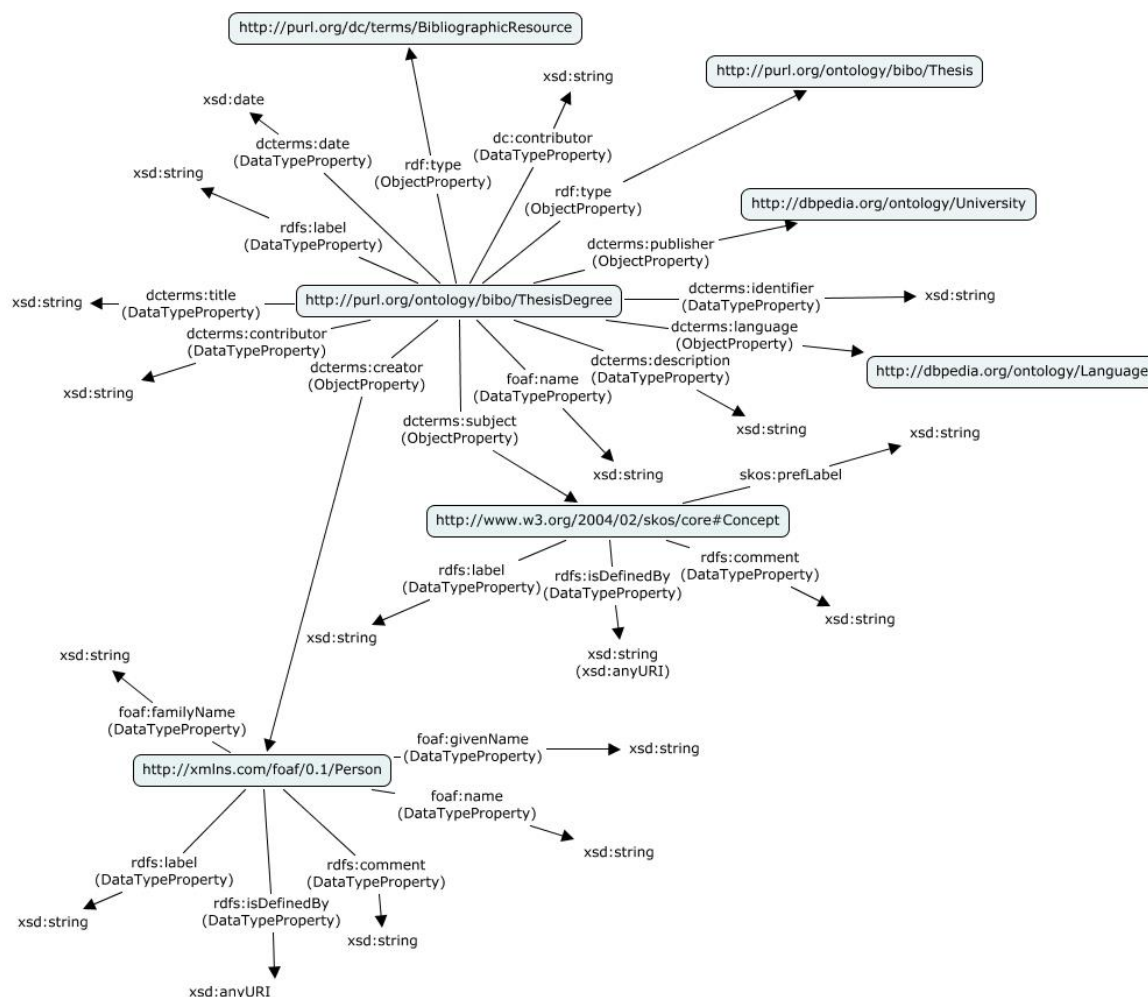


Figura 7. Grafo Vocabularios Tesis

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

⁸ <http://data.utpl.edu.ec/serendipity/oar>

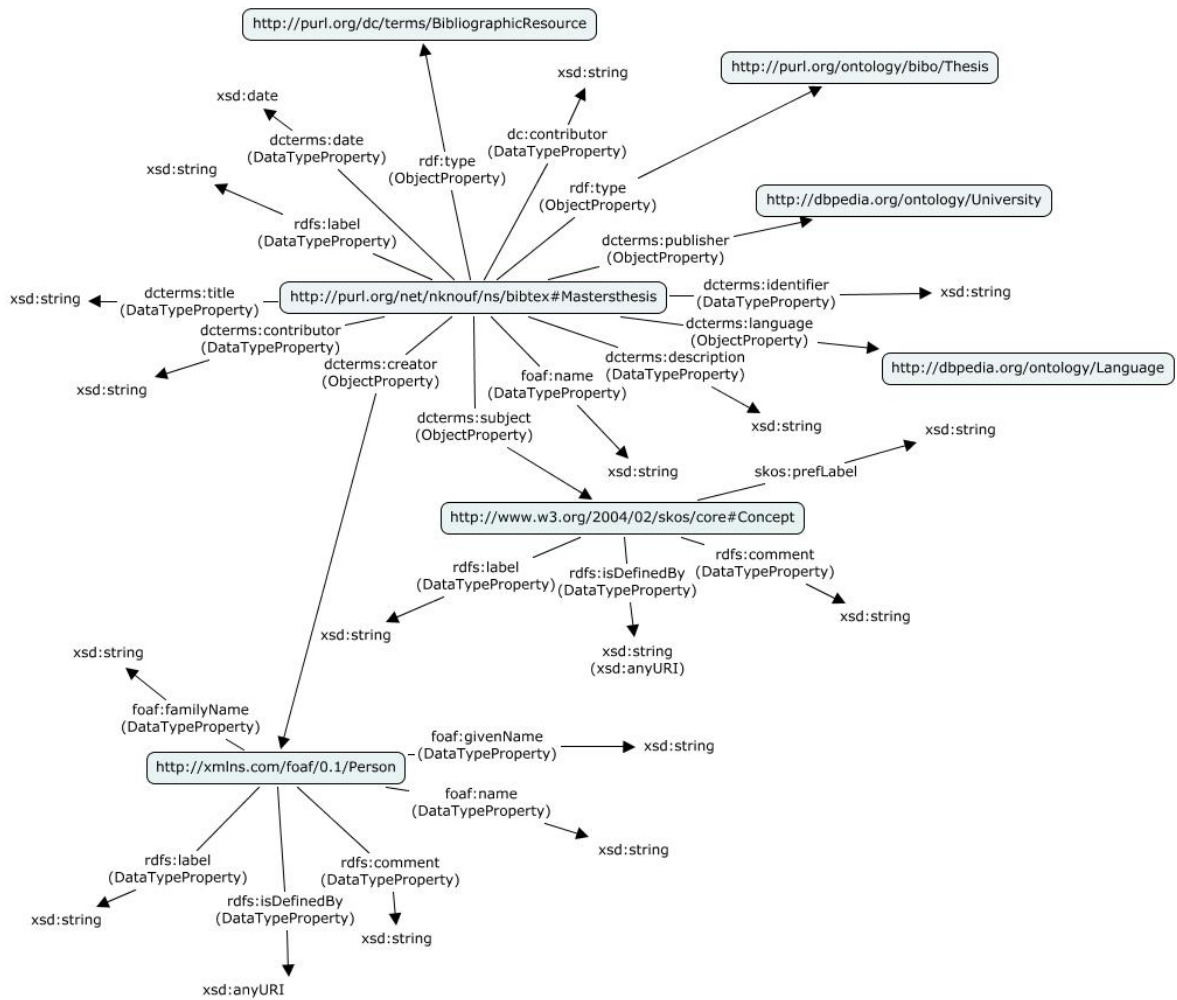


Figura 8. Grafo Vocabularios Maestría
 Fuente: Datos Serendipity
 Elaboración: Jonathan Yaguachi, Octubre 2015

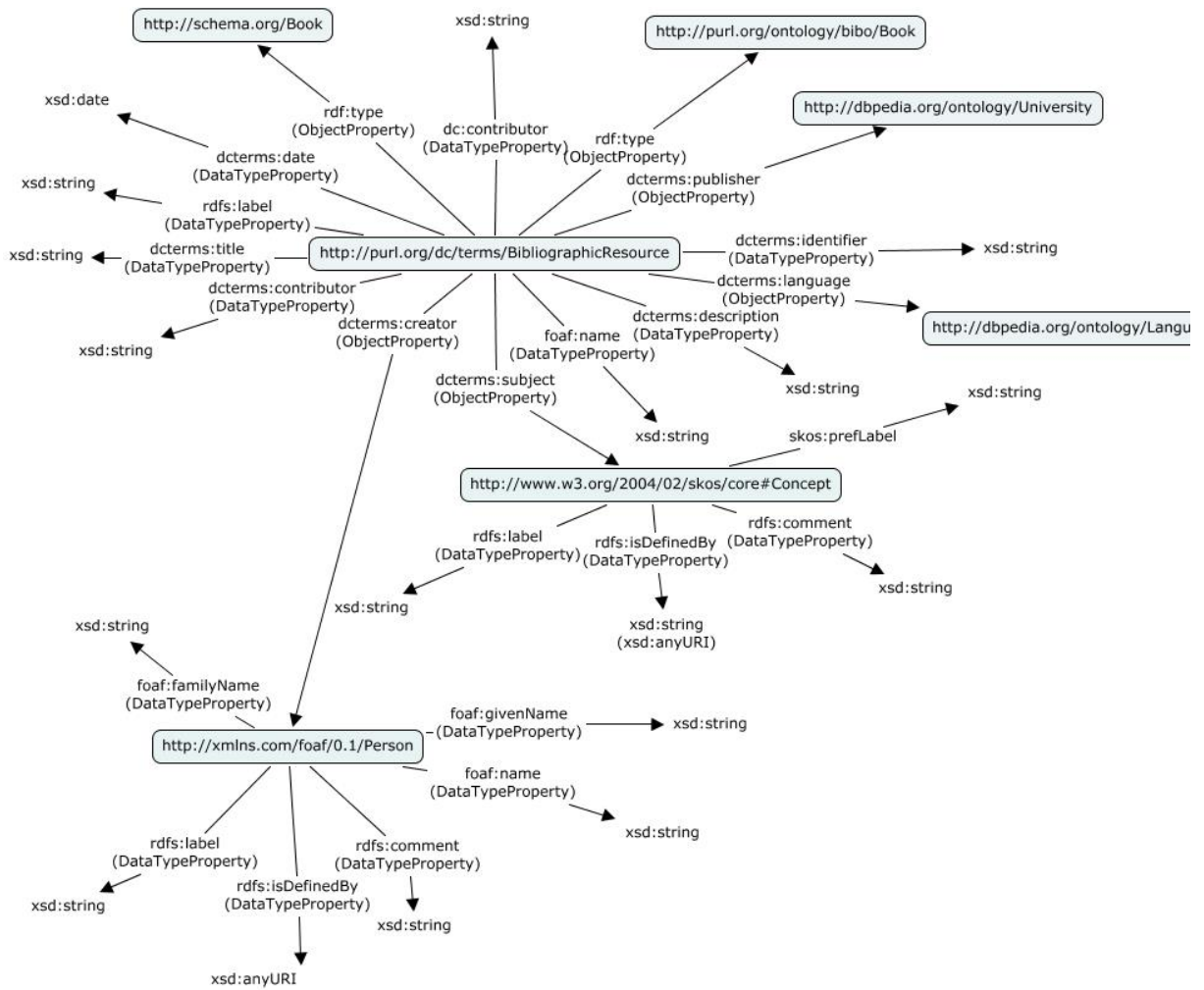


Figura 10. Grafo Vocabularios Libro.
Fuente: Datos Serendipity
Elaboración: Jonathan Yaguachi, Octubre 2015

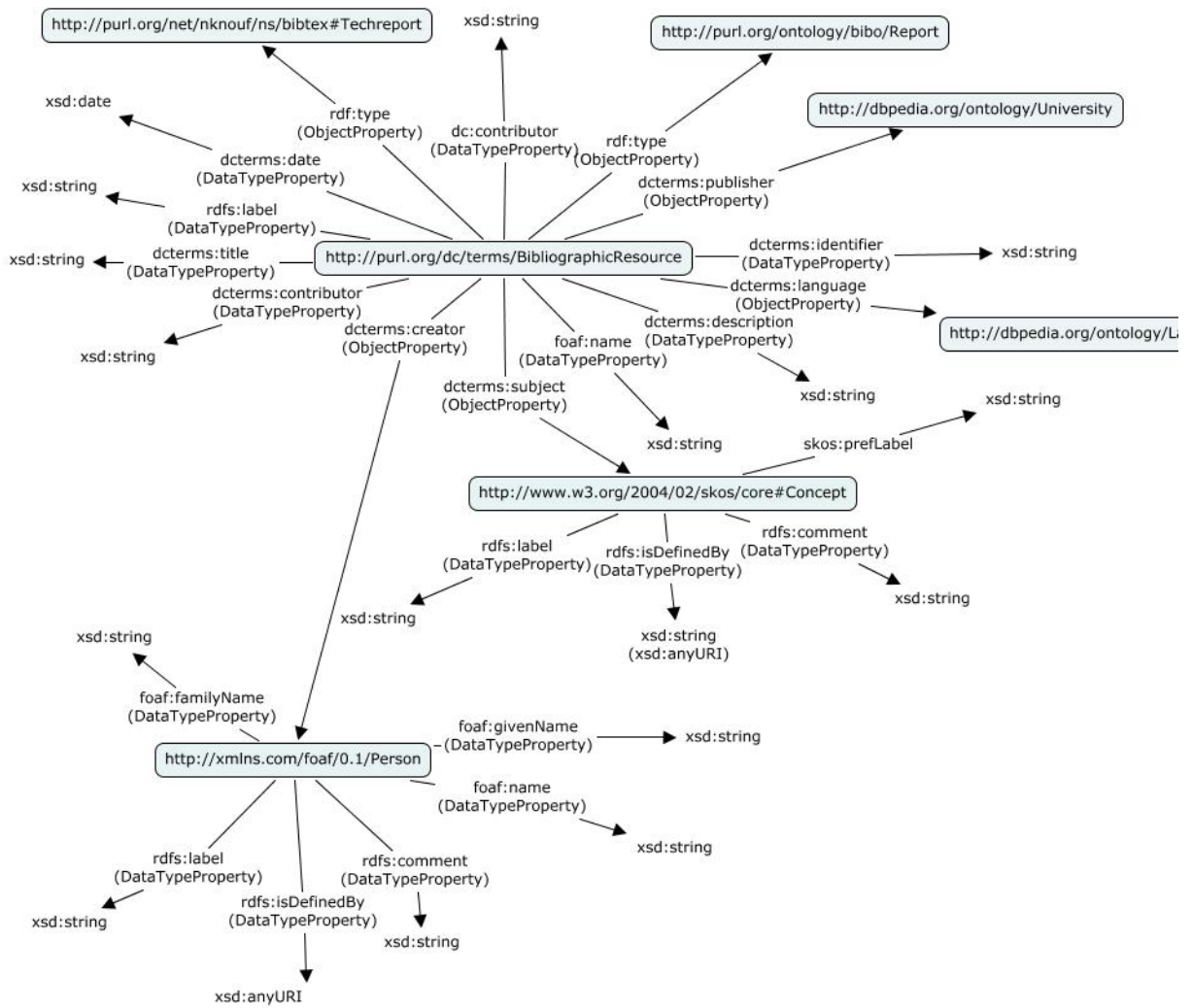


Figura 11. Grafo Vocabularios Reporte
Fuente: Datos Serendipity
Elaboración: Jonathan Yaguachi, Octubre 2015

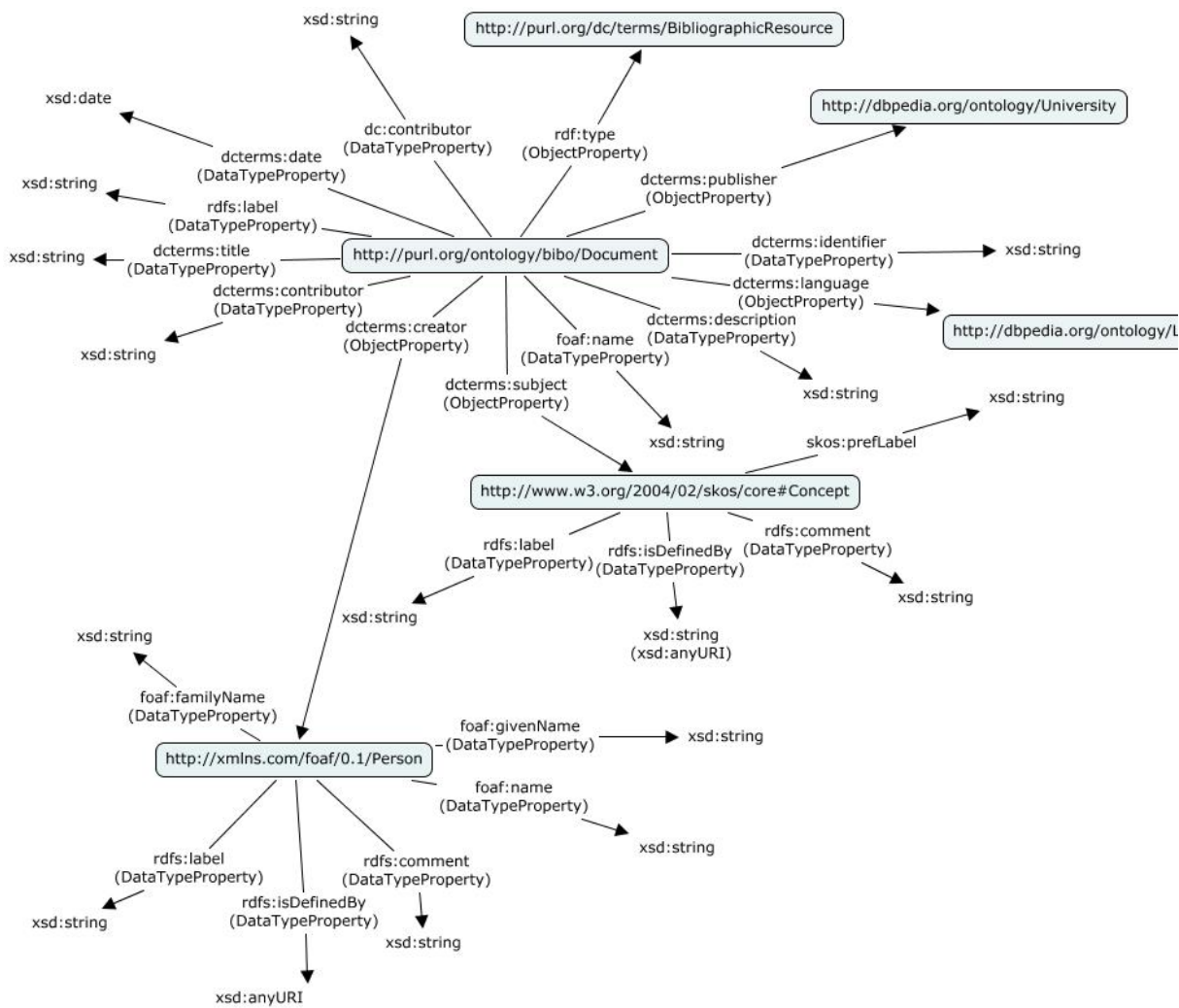


Figura 12. Grafo Vocabularios Documento
Fuente: Datos Serendipity
Elaboración: Jonathan Yaguachi, Octubre 2015

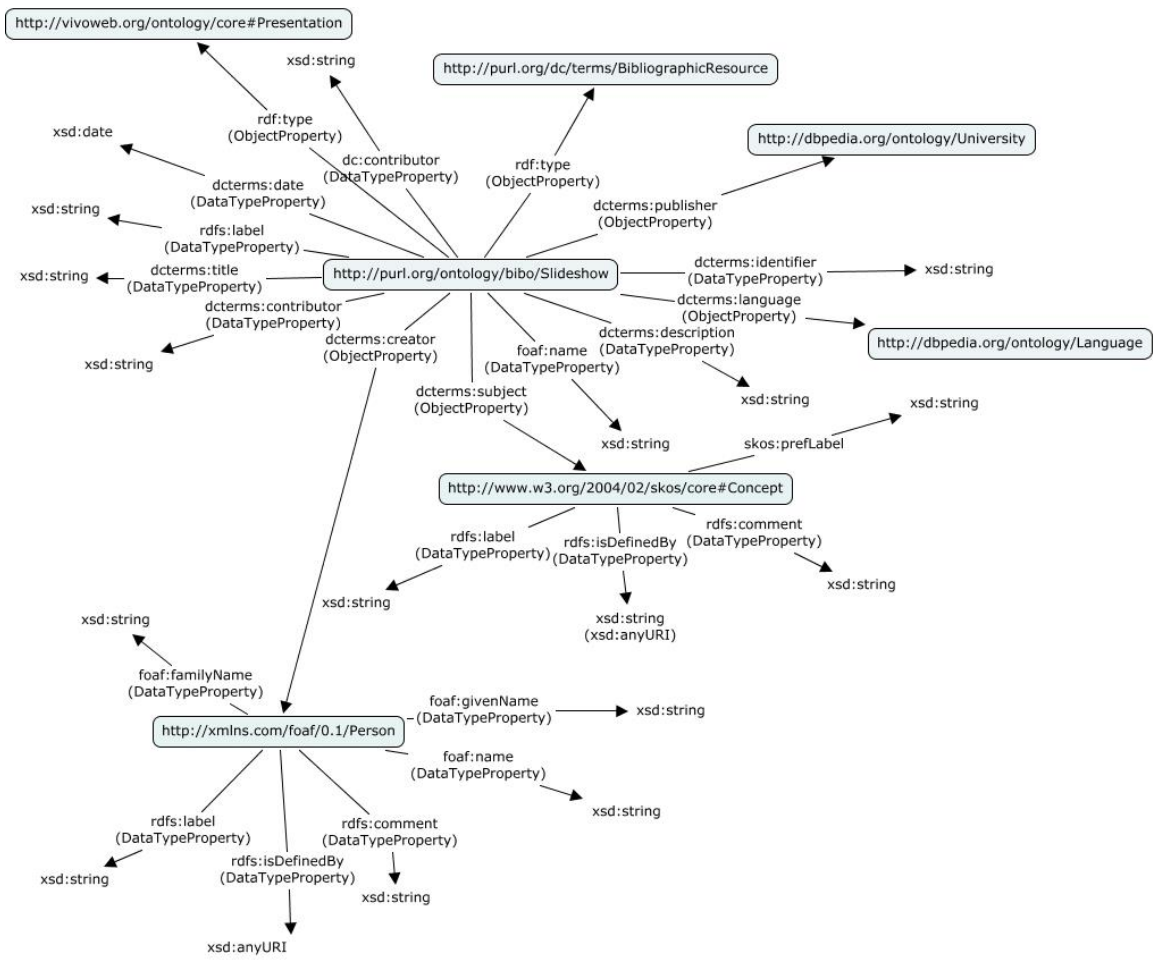


Figura 13. Grafo Vocabularios Presentación
Fuente: Datos Serendipity
Elaboración: Jonathan Yaguachi, Octubre 2015

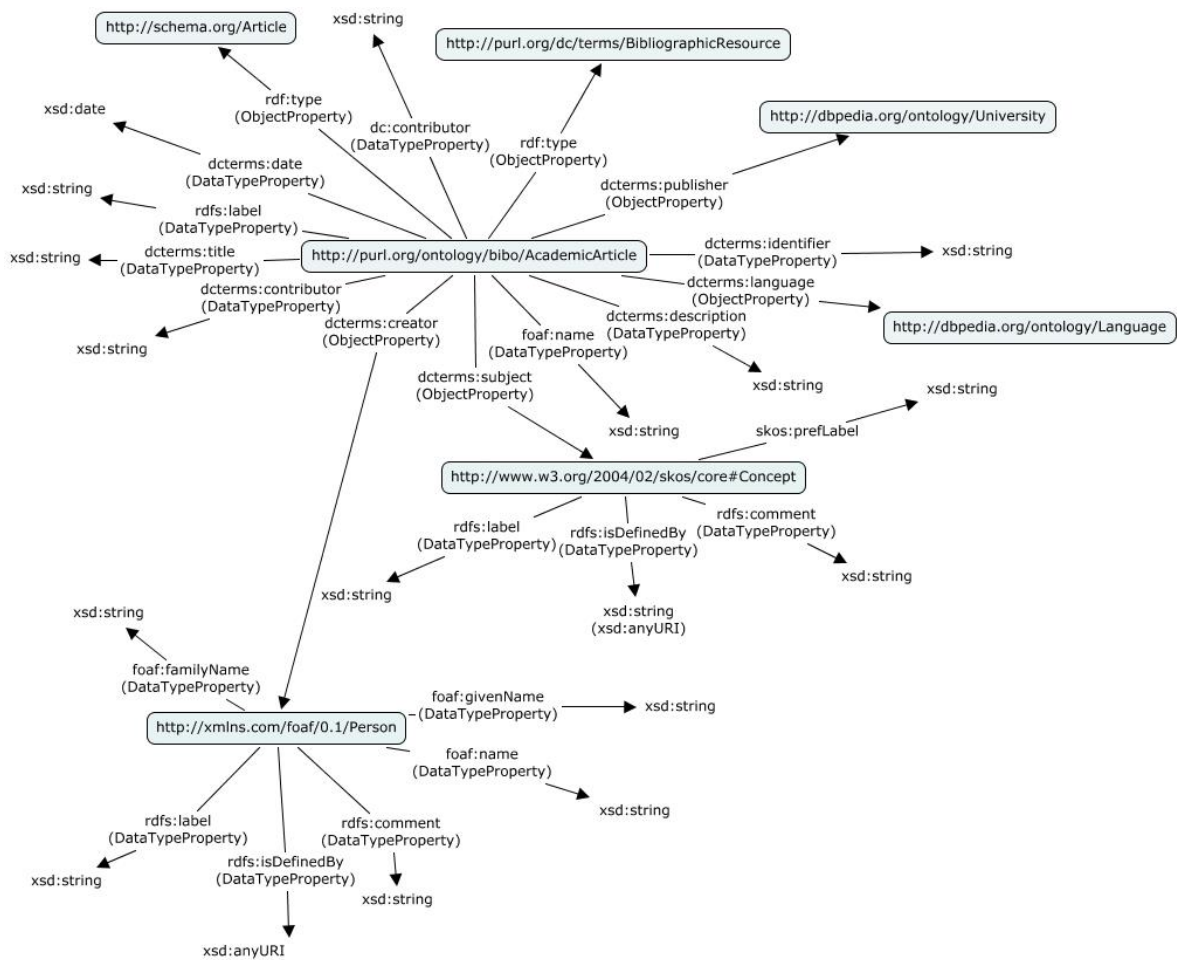


Figura 14. Grafo Vocabularios Artículo

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

CAPÍTULO III: SERVICIOS WEB

3.1 Introducción

En el presente capítulo se presenta la descripción del desarrollo e implementación de servicios web que permitirán consumir los datos tratados en el Capítulo

CAPÍTULO II: DESCRIPCIÓN DE DATOS, los mismos que se encuentran almacenados en Virtuoso. Estos datos se podrán visualizar utilizando alguna de las herramientas de visualización observadas anteriormente en la Tabla 3. Formato JSON de las herramientas seleccionadas.

EL objetivo de la creación de estos Servicio Web es la necesidad de obtener una aplicación suficientemente genérica en cada uno de sus módulos de tal manera que si posteriormente cambia alguno de ellos, el cambio solo afecte a este sin tener que realizar cambios en el resto de módulos, es decir que cada uno de los componentes que forman parte del proceso para la visualización tienen que estar creados de forma aislada pero a la misma vez listos para el intercambio de mensajes para la comunicación síncrona entre servicios lo que permite trabajar sobre los mismos datos.

Es así como se pretende crear servicios web que gestionen el flujo de los datos en módulos separados de tal forma que cada uno se encargue de una tarea en específico, para la solución de este planteamiento.

3.2 Arquitectura

Para un mejor entendimiento del funcionamiento de la aplicación en la Figura 16. Arquitectura Servicios Web que describe de forma fácil y sencilla la arquitectura bajo la cual trabaja la aplicación así como también cada uno de los elementos que intervienen para su ejecución.

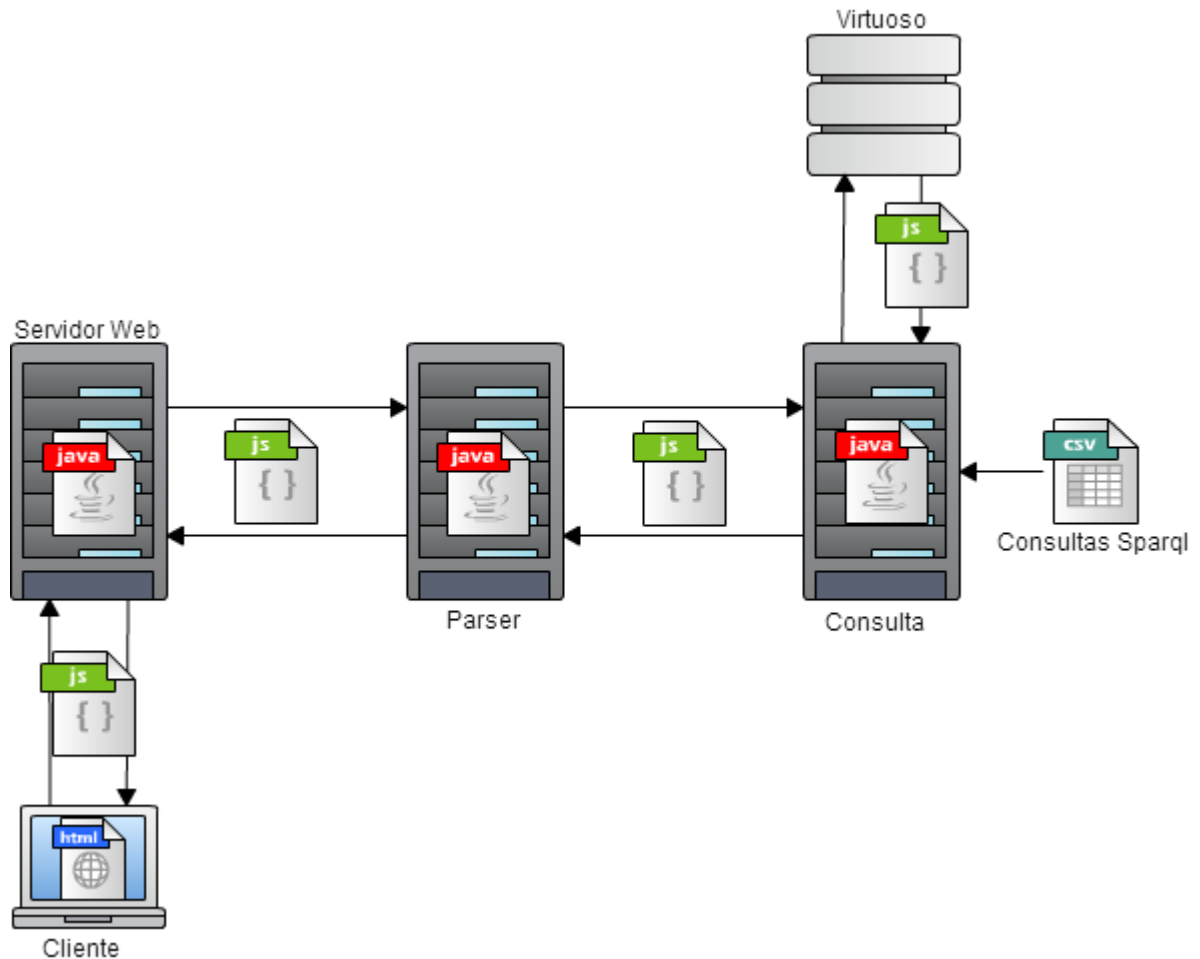


Figura 16. Arquitectura Servicios Web
 Fuente: Datos Serendipity
 Elaboración: Jonathan Yaguachi, Octubre 2015

En base a la arquitectura presentada en la Figura 16. Arquitectura Servicios Web a continuación se presenta la descripción de cada uno de los elementos usados en el diseño de la aplicación.

3.2.1 Virtuoso.

Triple Store cuya función será la de almacena los datos con los cuales se trabajara, es un servidor de consulta que contiene los datos semánticos de recursos educativos publicados por la Universidad Técnica Particular de Loja detallados en el capítulo

CAPÍTULO II: DESCRIPCIÓN DE DATOS.

Las consultas a este servidor se las realiza mediante SPARQL, un lenguaje estándar para la consulta de grafos, el resultado que se obtiene es un conjunto de datos en formato sujeto, predicado objeto.

En la Tabla 11. Numero de recursos Serendipity OAR se presenta una descripción rápida de los datos que se encuentran en virtuoso y los cuales se utilizaron para la elaboración de la aplicación.

Tabla 11. Numero de recursos Serendipity OAR

Descripción	Número
Número de Tripletas	174715
Número de Entidades	18287
Número de Clases	20
Número de Predicados	17
Número de Sujetos	18288
Número de Objetos	12658

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

Así mimos como ya se lo ha mencionado las consultas serán realizadas al EndPoint publicado en la dirección data.utpl.edu.ec/serendipity/oar/sparql usando el grafo data.utpl.edu.ec/serendipity/oar/, mediante consultas SPARQL y cuyo resultado será un conjunto de datos en formato JSON.

3.2.2 Consulta.

Servicio Web de Consulta, es el intermediario entre el Servicio Web y Virtuoso, así en el servicio se reciben la petición realizada por el usuario junto con los parámetros seleccionados, con esta información se extrae una platilla de consultas SPARQL almacenadas en un archivo CSV e identificadas por el tipo de petición realizada por el

usuario, se remplazan los parámetros por sus correspondientes y luego de esto se envían a Virtuoso para que la consulta sea ejecutada, el resultado es un conjunto de datos en formato JSON el mismo que es transformado a otra estructura JSON para obtener una estructura ordenada de los datos además de sus relaciones necesaria para la distribución en cada una de las herramientas de visualización

En la Tabla 12. Especificación de Caso de Uso, Consulta de Datos se describe de forma detallada cada uno de los elementos que intervienen en este servicio.

Tabla 12. Especificación de Caso de Uso, Consulta de Datos

Especificación del caso de uso: Consulta de Datos	
Código	1
Nombre	Consulta de Datos
Descripción	Este caso de uso describe el proceso de extracción de datos desde Virtuoso hacia la aplicación.
Actores	Usuario, Virtuoso, Aplicación.
Precondición	El usuario debe configurar parámetros de entrada como la ruta del grafo y el EndPoint
Post condición	Los parámetros de entrada deben de ser datos válidos y correctamente escritos para evitar problemas de comunicación con Virtuoso
Flujo normal	<ol style="list-style-type: none"> 1. Configurar parámetros de entrada 2. Extraer Plantilla Sparql 3. Reemplaza parámetros 4. Ejecuta consulta en Virtuoso 5. Aplicación extrae datos desde Virtuoso 6. Presenta resultado en formato JSON
Excepciones	Si el usuario no ingresa los parámetros correctamente, no se realizara ninguna acción por falta de datos.
Anotaciones	Ninguna

Fuente: Caso de Uso, Consulta de Datos

Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.2.1 Parámetros de Entrada.

Los parámetros de entrada son los datos a consultar en virtuoso por los general estos datos viajan en la URL.

En Jersey existen dos tipos de parámetros los cuales se clasifican como Path Parameters y Query Parameters.

Los Path Parameters son rutas relativas a la URL inicial, puede haber algunas de ellas en una la misma ruta. Ejemplo: una ruta normal puedes como esta www.localhost.com adicional a esto se puede enviar parámetros como parte de la ruta como el /país o la ciudad /ciudad la ruta quedaría así: www.localhost.com/Ecuador/Loja siendo Ecuador y Loja parámetros de entrada para el servicio web.

Los Query Parameters son parámetros que normalmente están asociados a peticiones GET y POST de HTTP, tienen una estructura clave – valor y pueden existir algunos de ellos en la misma ruta. Se los identifica en la ruta cuando se observa un carácter de cierre de pregunta “?” que indica que a continuación va ir una serie de parámetros de consulta, para separar entre varios de estos parámetros se una el carácter “&”.

Ejemplo: tomando como base el ejemplo anterior www.localhost.com/Ecuador/Loja?intitucion=SOLCA&user=username.

Para el usuario todo este proceso es transparente ya que no tiene que estar escribiendo todos estos parámetros en el navegador si no que lo realiza de forma dinámica escogiendo opciones desde una lista de valores.

3.2.2.2 Consultas SPARQL.

Las consultas Sparql se almacenan en un archivo CSV con el objetivo de que si algún punto es requerido modificar una o varias consultas poder hacerlo directamente en este archivo y no tener que modificar la aplicación.

En la Tabla 13. Consultas Sparql se presenta las consultas que actualmente están almacenadas en este archivo y las que se están usando para la ejecución de la aplicación.

Tabla 13. Consultas Sparql

1	<code>select distinct ?o as ?label where {?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?o . ?s ?p ?o} order by ?s</code>
2	<code>select distinct ?o as ?label, count(?o) as ?number where{ ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?o . ?s ?p ?o }</code>
3	<code>select distinct ?institution where { ?s <http://purl.org/dc/terms/publisher> ?institution }</code>
4	<code>select distinct ?institution where { \$1 <http://www.w3.org/2000/01/rdf-schema#label> ?institution . FILTER(LANG(?institution) = "es") }</code>

5	<pre>select ?s ?pre ?obj where{ { ?s ?pre ?obj . { select distinct ?s where { ?s <http://purl.org/dc/terms/publisher> <\$publisher> . ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <\$type> . ?s ?p ?o. filter regex (?o, "\$search", "i") } } } union { select <\$publisher> as ?s ?pre ?obj where { <\$publisher> ?pre ?obj . } } } order by ?s</pre>
6	<pre>select ?s ?pre ?obj where{ { select <\$publisher> as ?s ?pre ?obj where { <\$publisher> ?pre ?obj . } } union { ?s ?pre ?obj . { select distinct ?s where { ?s <http://purl.org/dc/terms/publisher> <\$publisher> . ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <\$type> . ?s ?p ?o. filter regex (?o, "\$search", "i") } } } union { ?su <http://purl.org/dc/terms/publisher> <\$publisher> . ?su <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <\$type> . ?su <http://purl.org/dc/terms/creator> ?s. ?s ?pre ?obj filter regex (?s, "\$search", "i") } } } order by ?s</pre>

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

El archivo CSV está constituido básicamente por dos columnas la primera de ella es un secuencial numérico el cual será de ayuda para identificar a cada consulta dentro del archivo, por lo tanto este número no puede repetirse; la segunda columna es la consulta sparql.

Para evitar problemas con delimitador o separador del archivo CSV con algunos caracteres que usan las consultas sparql se cambió el delimitador por un “%” como lo muestra a continuación la Figura 17. Archivo queries.csv

1%select distinct ?o as ?label where { ?s <http
2%select distinct ?o as ?label, count(?o) as ?number where { ?s <http
3%select distinct ?institution where { ?s <http
4%select distinct ?institution where { \$1 <http
5%select ?s ?pre ?obj where{ { ?s ?pre ?obj . { select distinct ?s where { ?s <http
6%select ?s ?pre ?obj where{ { select <\$publisher> as ?s ?pre ?obj where { <\$publisher> ?pre ?obj . } } } union { ?s ?pre ?obj . { select distinct ?s where { ?s <http

Figura 17. Archivo queries.csv⁹

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

Las consultas Sparql que se encuentran en el archivo queries.csv representa a la estructura de una consulta sparql ya que poseen variables a ser remplazadas según los parámetros de consulta escogidos por el usuario; los parámetros son los mismos que envían en la URL de solicitud de algún servicio web.

⁹ <http://www.convertcsv.com/csv-viewer-editor.htm>

Por ejemplo según el archivo de consultas presentado anteriormente se escoge la consulta número 5 la consulta es la que se muestras en la Figura 18. Ejemplo Consulta Sparql.

```
select ?s ?pre ?obj
where {
  {
    ?s ?pre ?obj .
    {
      select distinct ?s
      where {
        ?s <http://purl.org/dc/terms/publisher> <$publisher> .
        ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <$type> .
        ?s ?p ?o.
        filter regex (?o, "$search", "i")
      }
    }
  }
  union
  {
    select <$publisher> as ?s ?pre ?obj
    where {
      <$publisher> ?pre ?obj .
    }
  }
}
order by ?s
```

Figura 18. Ejemplo Consulta Sparql

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

En esta consulta existen variables cuyo valor necesita ser remplazado por su valor correspondiente ya que de otra manera su ejecución no devolverá valor o saldrá un error al momento de ejecutar; las variables a ser remplazadas se las puede identificar en la consulta porque a estas les preceden el símbolo dólar "\$". En la consulta de ejemplo existen tres variable a las cuales remplazar \$publisher, \$type y \$search; aunque \$publisher aparece en tres ocasiones representa a la misma variable.

Los valores a remplazar son tomados de la URL de petición, a continuación se presenta un ejemplo de URL con el valor de los parámetros a remplazar.

http://carbono.utpl.edu.ec:8080/VDEOers-1.0-SNAPSHOT/webresources/vde/params?graph=1&publisher=http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja&type=http://purl.org/ontology/bibo/Thesis&search=Cabrera

Tabla 14. Remplazo de Parámetros Consultas Sparql.

Parámetro	Valor
\$publisher	http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja
\$type	http://purl.org/ontology/bibo/Thesis
\$search	Cabrera

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Una vez reemplazado los parámetros la consulta sparql tiene como resultado como se muestras en la Figura 19. Ejemplo Consulta Sparql, Reemplazo de parámetros.

```

select ?s ?pre ?obj
where {
  {
    ?s ?pre ?obj .
    {
      select distinct ?s
      where {
        ?s <http://purl.org/dc/terms/publisher> <http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja> .
        ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/bibo/Thesis> .
        ?s ?p ?o.
        filter regex (?o, "Cabrera", "i")
      }
    }
  }
  union
  {
    select <http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja> as ?s ?pre ?obj
    where {
      <http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja> ?pre ?obj .
    }
  }
}
order by ?s

```

Figura 19. Ejemplo Consulta Sparql, Reemplazo de parámetros

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

El resultado de esta consulta es ejecutada en el EndPoint de Virtuoso publicado por la Universidad Técnica Particular de Loja, devuelve los datos consultados en formato JSON como lo muestra en la Figura 20. Resultado EndPoint Virtuoso



Figura 20. Resultado EndPoint Virtuoso
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

El JSON presentado anteriormente es una estructura estándar es decir cualquiera que sea la consulta ejecutada en Virtuoso la respuesta siempre va a seguir la misma estructura; un objeto *head* que contiene un arreglo *link* y otro arreglo *vars* con las variables usadas en la consulta sparql; adicionalmente también se tiene el objeto *results* que trae algunos detalles de la consulta ejecutada y además el arreglo *bindings* que bastamente son los datos de resultado de la consulta.

La aplicación realizará una conversión de esta estructura JSON a otra la cual resulta más práctica a la hora de identificar los nodos de información y las relaciones que estos poseen la Figura 21. Conversión de JSON muestra cómo será el resultado de esta conversión.

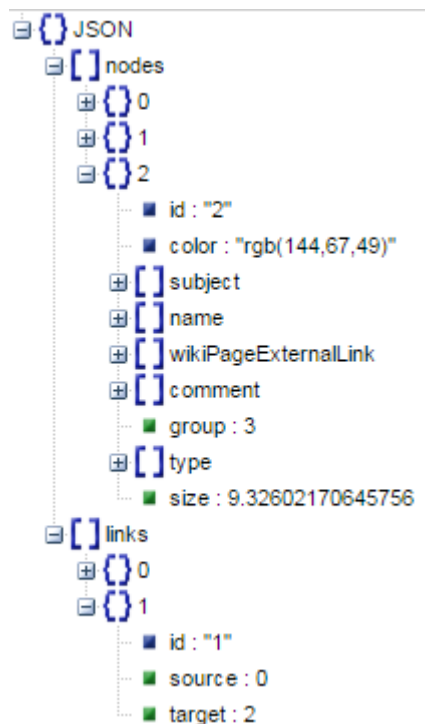


Figura 21. Conversión de JSON
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

A continuación se presenta cada uno de los servicios disponibles en el módulo de Consulta.

3.2.2.3 Obtención de Instituciones.

Permite obtener un listado de Instituciones de entre las cuales exista recursos educativos abiertos registrados en el repositorio Serendipity OAR¹⁰ y que se haya hecho la extracción en el EndPoint de Virtuoso¹¹. El detalle de este servicio se lo encuentra en la Tabla 15. Descripción Obtención de Instituciones.

Tabla 15. Descripción Obtención de Instituciones.

Descripción	Obtención de Instituciones
URL de Solicitud	.../VDEOers-1.0-SNAPSHOT/webresources/vde
Tipo de Petición:	GET
Formato de Respuesta	JSON
Parámetros	--

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

¹⁰ <http://data.utpl.edu.ec/serendipity/oar>

¹¹ <http://data.utpl.edu.ec/serendipity/oar/sparql>

Ejemplo:

GET .../VDEOers-1.0-SNAPSHOT/webresources/vde

Resultado: Figura 22. JSON de Obtención de Institución.

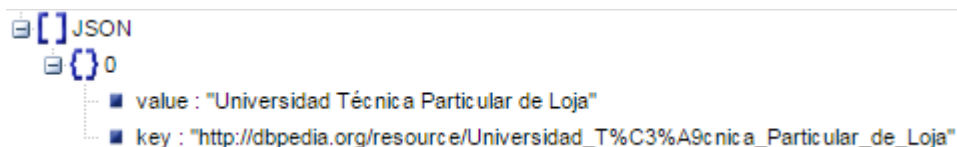


Figura 22. JSON de Obtención de Institución.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.2.4 Obtención de Tipos de Recursos.

Obtiene un listado de los tipos de recursos educativos abiertos registrados en el repositorio Serendipity OAR. Esta petición es dependiente de la institución seleccionada previamente de tal forma que si se escoge como institución la Universidad Técnica Particular de Loja mostrara todos los tipos de recursos disponibles para esta institución. El detalle de este servicio se lo encuentra en la Tabla 16. Descripción de Obtención de Tipos de Recursos

Tabla 16. Descripción de Obtención de Tipos de Recursos.

Descripción	Obtención de Tipos de Recursos
URL de Solicitud	.../VDEOers-1.0-SNAPSHOT/webresources/vde/type
Tipo de Petición:	GET
Formato de Respuesta	JSON
Parámetros	--

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Ejemplo:

GET .../VDEOers-1.0-SNAPSHOT/webresources/vde/type

Resultado: Figura 23. JSON de Obtención de Tipos de Recursos.

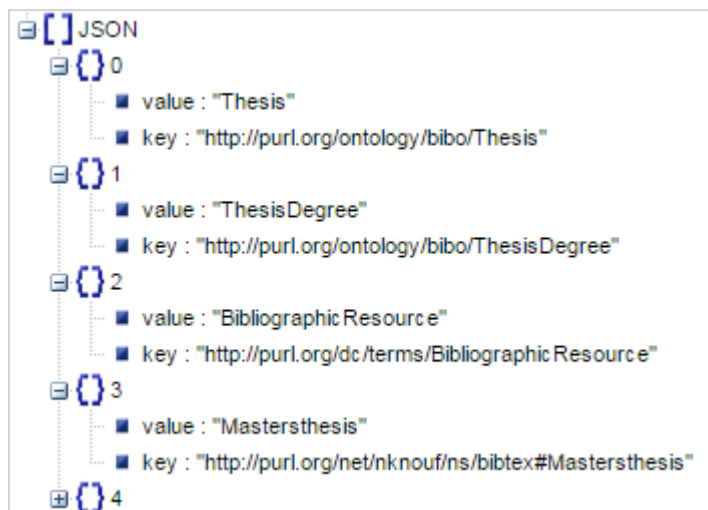


Figura 23. JSON de Obtención de Tipos de Recursos.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.2.5 Obtención de OERs.

Obtiene un listado de los recursos educativos abiertos asociadas con la institución y al tipo de recurso seleccionados previamente, se mostrara los OERs que se encuentren registrados en dentro del repositorio Serendipity OAR. El detalle de este servicio se lo encuentra en la Tabla 17. Obtención de OERs.

Tabla 17. Obtención de OERs.

Descripción	Obtención de OERs	
URL de Solicitud	.../VDEOers-1.0-SNAPSHOT/webresources/vde/params	
Tipo de Petición:	GET	
Formato de Respuesta	JSON	
Parámetros	graph (Obligatorio)	Representa el tipo de grafico a presentar, es un parámetro numérico y cada valor que contiene identifica a cada una de las gráficas disponibles para la visualización. Ejemplo: 1
	publisher (Obligatorio)	Corresponde a la Institución en la cual se pretende realizar la búsqueda su valor corresponde a la URI que describa a la institución dentro del repositorio Serendipity

		OAR. Ejemplo: http://es.dbpedia.org/page/Universidad_T%C3%A9cnica_Particular_de_Loja
	type (Obligatorio)	Identifica a un Tipo de Recurso Educativo Abierto disponible en el repositorio Serendipity OAR, entre los principales se puede mencionar a Tesis o Maestrías. Su Ejemplo: http://purl.org/ontology/bibo/ThesisDegree
	search (Obligatorio)	Parámetro que se usa para filtrar datos en función a una cadena de texto ingresada manualmente, esta cadena de texto no es sensitiva a mayúsculas pueden ir en con mayúsculas o en minúsculas. Ejemplo: Cabrera

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Ejemplo:

GET .../VDEOers-1.0-SNAPSHOT/webresources/vde/params?graph=1&publisher=http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja&type=http://purl.org/ontology/bibo/ThesisDegree&search=Cabrera

Resultado: Figura 24. Obtención de OERs.

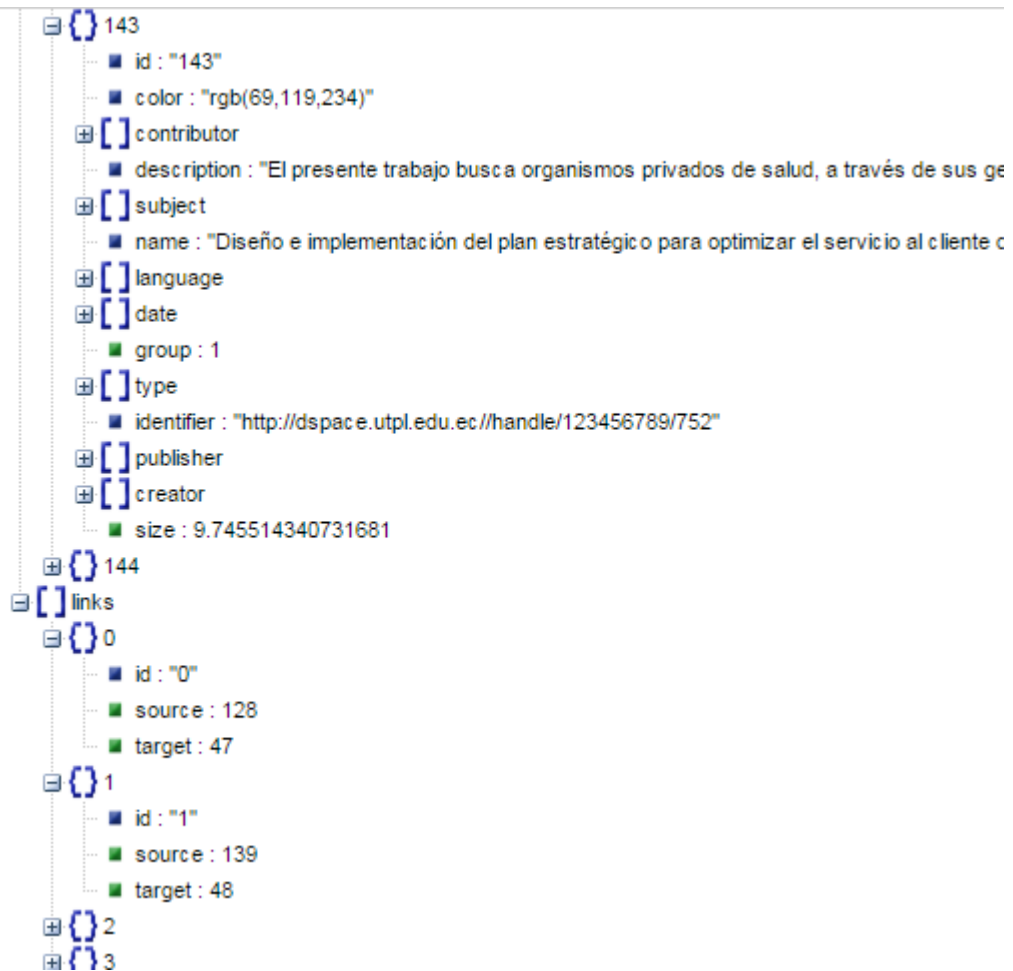


Figura 24. Obtenci\u00f3n de OERs.
 Fuente: Servicios Web VDE
 Elaboraci\u00f3n: Jonathan Yaguachi, Octubre 2015

Cada vez que se realicen peticiones a este servicio, este siempre devolver\u00e1 los datos en formato JSON. El JSON que se devuelve en un conjunto de datos est\u00e1ndar es decir que son un conjunto de datos que servir\u00e1 como base para armar otro JSON a partir de este, dependiendo de la herramienta de Visualizaci\u00f3n que lo requiera por lo cual la estructura del mismo consiste en establecer los respectivos nodos y sus relaciones. Un ejemplo m\u00e1s claro de esta estructura se lo puede observar en la Figura 25. Estructura JSON General.

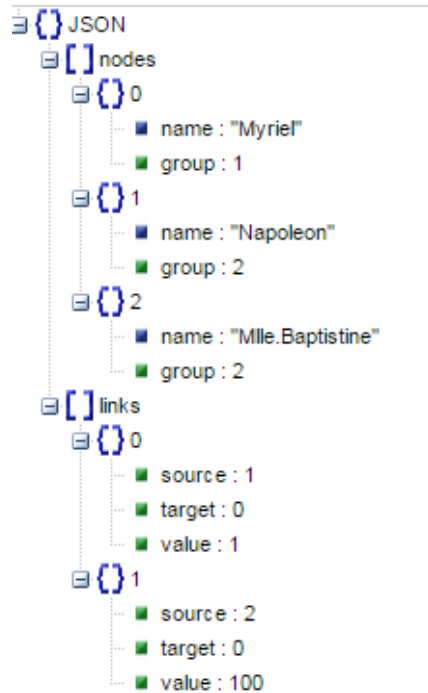


Figura 25. Estructura JSON General.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

Los nodos representan un objeto en específico, por ejemplo en el caso de los datos encontrados en el repositorio Serendipity OAR un nodo puede representar un Autor, una Institución o un Oer, dichos objetos a su vez tienen ciertas propiedades que lo representan en el caso del Autor se podrían tener las propiedades identificación, nombre, apellido, edad, etc. Las propiedades varían dependiendo del Objeto, se puede decir que un determinado Autor tiene identificación, nombre, apellido, edad; mientras que otro solo puede contener identificación, nombre, apellido así como lo muestra la Figura 26. Descripción Nodos. Esto depende mucho de los datos que se encuentra en el EndPoint.

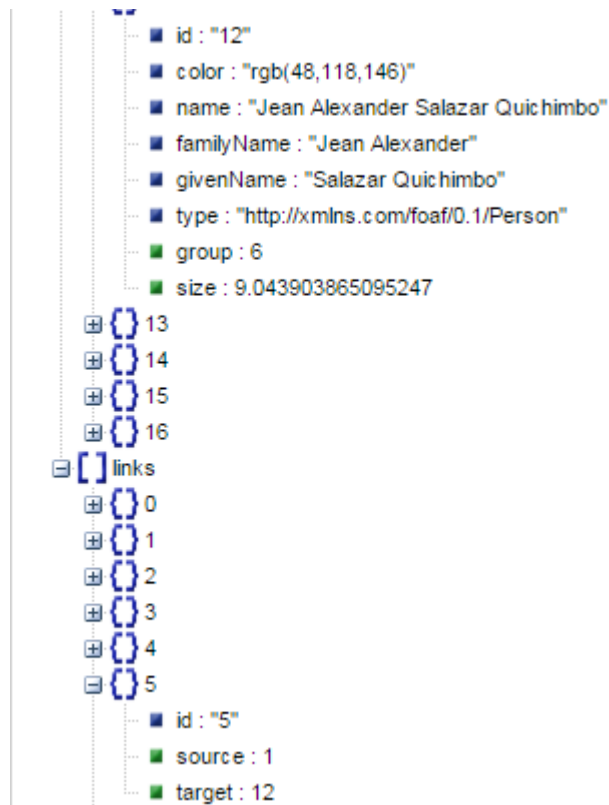


Figura 26. Descripción Nodos.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

Las relaciones es la parte donde determina que nodo va relacionado con otro, se utiliza un identificador que suele estar implícitamente por el orden de aparición de los nodos en la estructura JSON, entonces la relación consiste en ubicar dos elementos origen y destino que en el JSON se puede ver como *source* y *target*.

- Source, esta etiqueta describe o indica desde que nodo se quiere realizar la relación.
- Target, esta etiqueta describe o indica hacia que nodo se quiere realizar la relación.

El identificador del nodo en la estructura JSON, normalmente es un número secuencial iniciando desde cero (0) hasta el número de nodos existentes. En la Figura 27. Descripción Relaciones (Links) se observan los elementos que son parte de la relación.

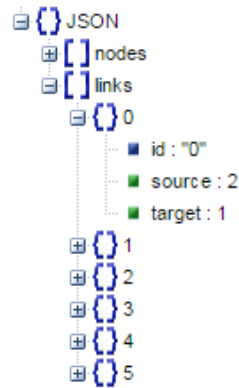


Figura 27. Descripción Relaciones (Links)

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Adicionalmente también se cuenta con un desarrollo que permite cambiar el nombre de la etiqueta del JSON generado, esto debido a que existe la posibilidad de querer cambiar el nombre de esta etiqueta por una que la describa mejor o simplemente para mayor comprensión a la hora de analizar el JSON.

Para parametrizar estos datos, la aplicación lee un archivo de equivalencias que precisamente esta en formato JSON en donde se especifican el valor por defecto con el que suele venir el dato, normalmente es el valor del predicado asociado a cada objeto dentro de cada tripleta, frente a este dato se ubica la etiqueta equivalente con el cual se quiere reemplazar este dato, la Figura 28. Archivo de Equivalencias muestra el archivo de equivalencias utilizado en la aplicación.

```

{
  "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [],
  "group": [],
  "color": [],
  "size": [],
  "id": [],
  "http://www.w3.org/2000/01/rdf-schema#label": ["titulo"],
  "http://www.w3.org/2004/02/skos/core#prefLabel": [],
  "http://purl.org/dc/terms/title": ["name"],
  "http://purl.org/dc/terms/creator": [],
  "http://purl.org/dc/terms/subject": [],
  "http://xmlns.com/foaf/0.1/familyName": ["Apellidos"],
  "http://xmlns.com/foaf/0.1/givenName": ["Nombres"],
  "http://xmlns.com/foaf/0.1/depiction": [],
  "http://xmlns.com/foaf/0.1/name": [],
  "http://purl.org/dc/terms/description": [],
  "http://purl.org/dc/terms/publisher": [],
  "http://purl.org/dc/terms/identifier": [],
  "http://purl.org/dc/terms/date": [],
  "http://purl.org/dc/elements/1.1/contributor": [],
  "http://purl.org/dc/terms/contributor": [],
  "http://purl.org/dc/terms/language": []
}

```

Figura 28. Archivo de Equivalencias
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

El archivo de equivalencias es un archivo JSON clave –valor en donde la clave es la el predicado de la tripleta y el valor es la etiqueta con la cual se quiere reemplazar, en los casos que no se encuentra valor y solo unos corchetes vacíos significa que el valor por defecto se mantiene y no será reemplazado por ningún valor adicional ver Figura 29. Cambio de Etiqueta JSON en donde se muestra el resultado una ejecución de prueba de cambio de etiqueta.

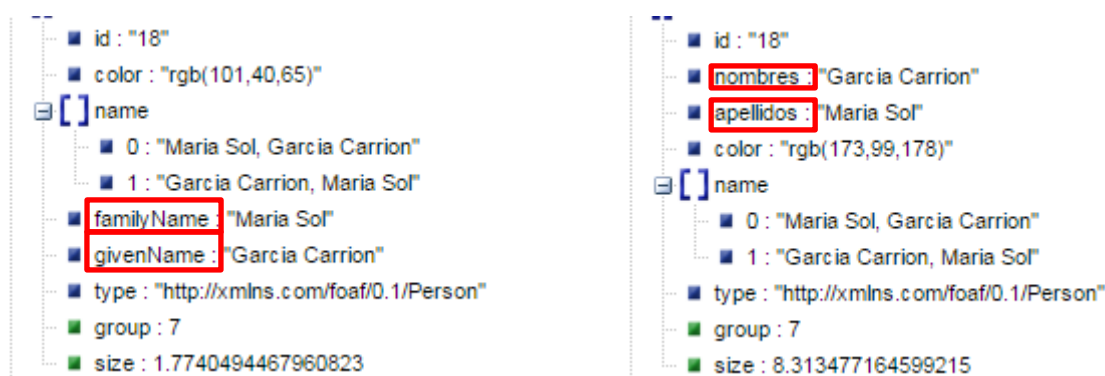


Figura 29. Cambio de Etiqueta JSON
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.2.6 Obtener Número de Oers por Tipo de Recurso.

Permite obtener un listado con el detalle de todos los tipos de recursos educativos abiertos que existen registrados en el EndPoint de Virtuoso juntamente con el número de recursos registrados por cada uno de ellos.

El resultado es un JSON con los datos de cada tipo de recurso y el número de recurso registradas por cada uno de ellos. El detalle de este servicio se lo encuentra en la Tabla 18. Descripción Obtención del Número de Oers por Tipo de Recurso.

Tabla 18. Descripción Obtención del Número de Oers por Tipo de Recurso.

Descripción	Obtener Número de Publicaciones por País.
URL de Solicitud	.../VDEOers-1.0- SNAPSHOT/webresources/vde/resourceType
Tipo de Petición:	GET
Formato de Respuesta	JSON
Parámetros	--

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Ejemplo:

.../VDEOers-1.0-SNAPSHOT/webresources/vde/resourceType

Resultado: Figura 30. JSON de Obtención del Número de Oers por Tipo de Recurso.

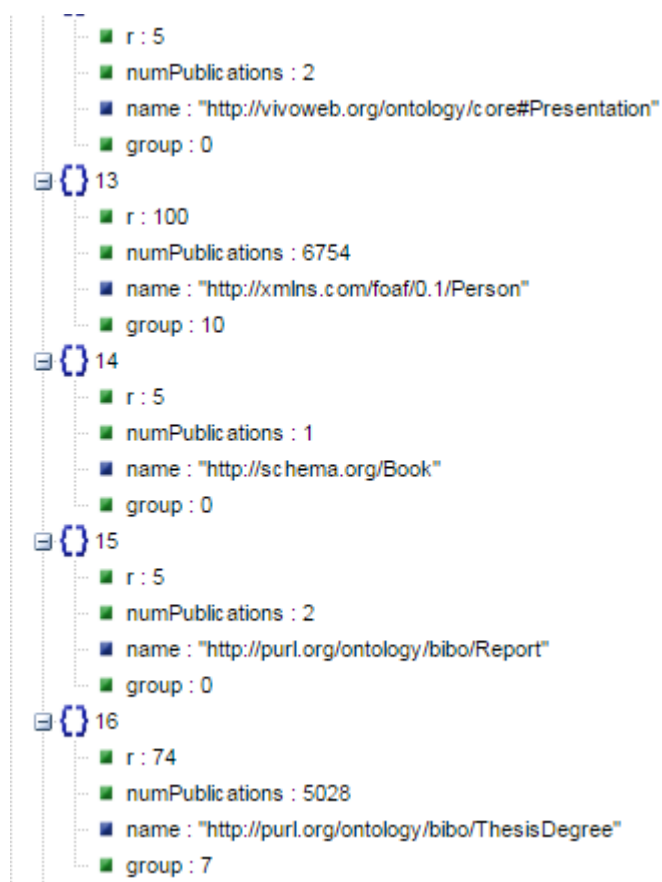


Figura 30. JSON de Obtención del Número de Oers por Tipo de Recurso.

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.3 Parser.

Servicio Web intermediario entre el Servicio de Consulta y el Servidor Web que realiza la conversión de estructura JSON, su función principal es la de hacer peticiones hacia el Servicio de Consulta obteniendo un JSON base el cual es transformado en una nueva estructura JSON en función al tipo de grafica a visualizar.

El servidor web solicita el JSON para una determinada herramienta de visualización, el trabajo del Servicio de Parser consiste en determinar que estructura JSON corresponde a

esa petición y formar dicha estructura y dar respuesta según los parámetros establecidos inicialmente.

En la Tabla 19. Especificación de Caso de Uso, Servicio Web Parser se describe de forma detallada cada uno de los elementos que intervienen en este servicio.

Tabla 19. Especificación de Caso de Uso, Servicio Web Parser

Especificación del caso de uso: Servicio Web Parser	
Código	2
Nombre	Parser
Descripción	Este caso de uso describe el proceso que se realiza para mapear la estructura JSON que se tiene originalmente y convertirla a otra en función de la gráfica seleccionada.
Actores	WS Consulta, WS Parser.
Precondición	El usuario debe configurar parámetros de entrada como la ruta del grafo y EndPoint
Post condición	Los parámetros de entrada deben de ser datos válidos y correctamente escritos para evitar problemas de comunicación con Virtuoso
Flujo normal	<ol style="list-style-type: none"> 1. Configurar parámetros de entrada 2. Consulta WS de Consulta 3. Validar tipo de Grafica 4. Arma JSON 5. Presenta resultado en formato JSON
Excepciones	Si el usuario no ingresa los parámetros correctamente, no se realizara ninguna acción por falta de datos.
Anotaciones	Ninguna

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.3.1 Parámetros de Entrada.

A igual que en el servicio web de consulta los parámetros de estrada son los mismos que ingresan en la URL de petición de recursos, la diferencia principal se tiene en el Path Parameter que cambias en lugar de **vde** se tiene **parser**, con el objetivo de identificar cual servicio es el que se está ejecutando.

Ejemplo:

http://localhost:8080/vdeoers/webresources/parser/params?graph=1&publisher=http://dbpedia.org/resource/Universidad_T%C3%A9cnica_Particular_de_Loja&type=http://purl.org/ontology/bibo/Thesis&search=Carrion

La tarea principal que realiza este servicio es consultar al servicio web de Consulta y obtener un JSON siempre con la misma estructura y convertir este según la estructura JSON de la gráfica que se requiere. En la Figura 31. Conversión entre JSON se presenta un ejemplo de conversión entre el JSON base a la estructura JSON requerido para la gráfica número 3 que se muestra en la Tabla 20. Código de Grafica y grafica a la cual representan

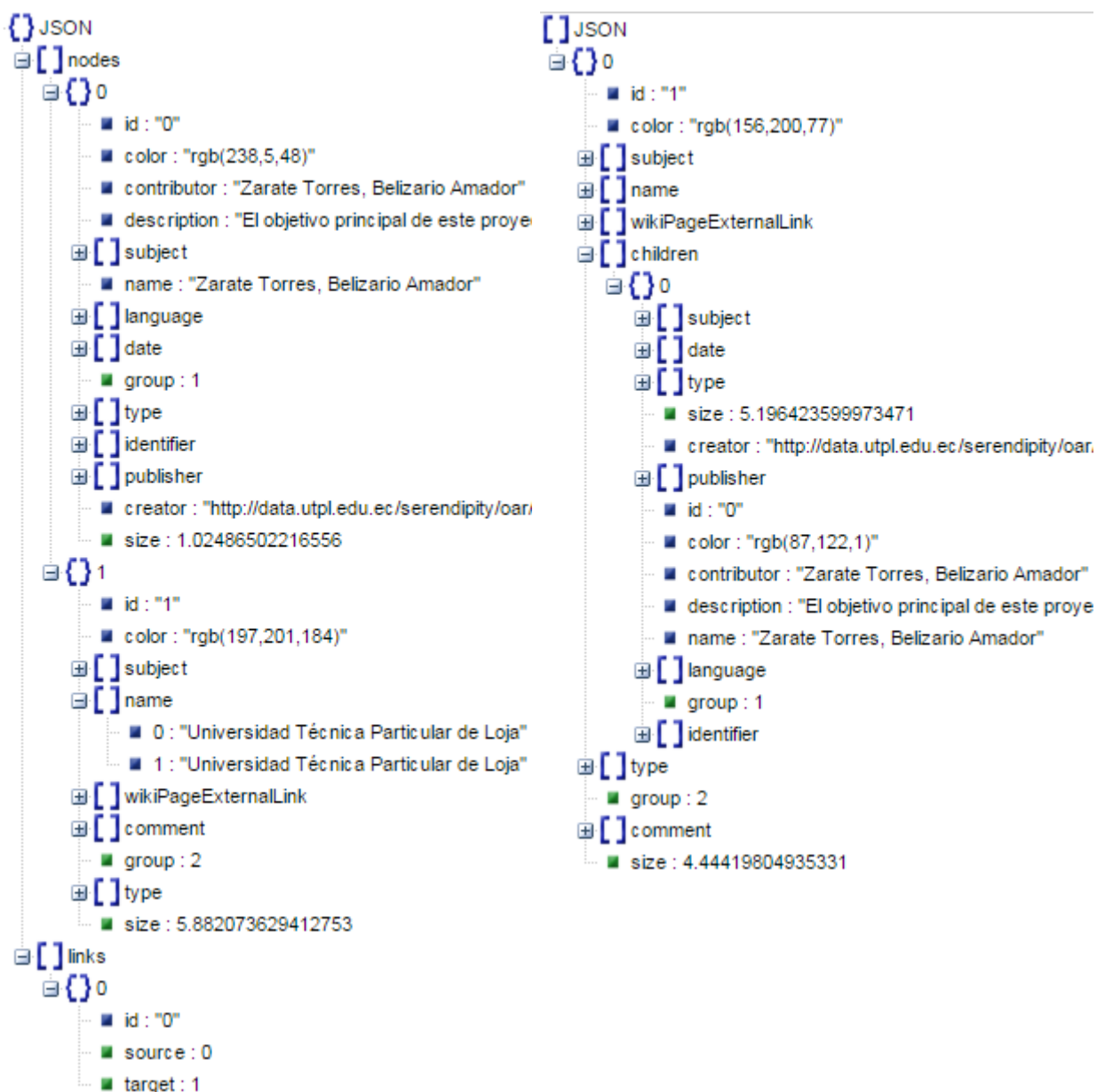


Figura 31. Conversión entre JSON
Fuente: Servicios Web VDE

A continuación se detalla los servicios disponibles en el servicio web Parser:

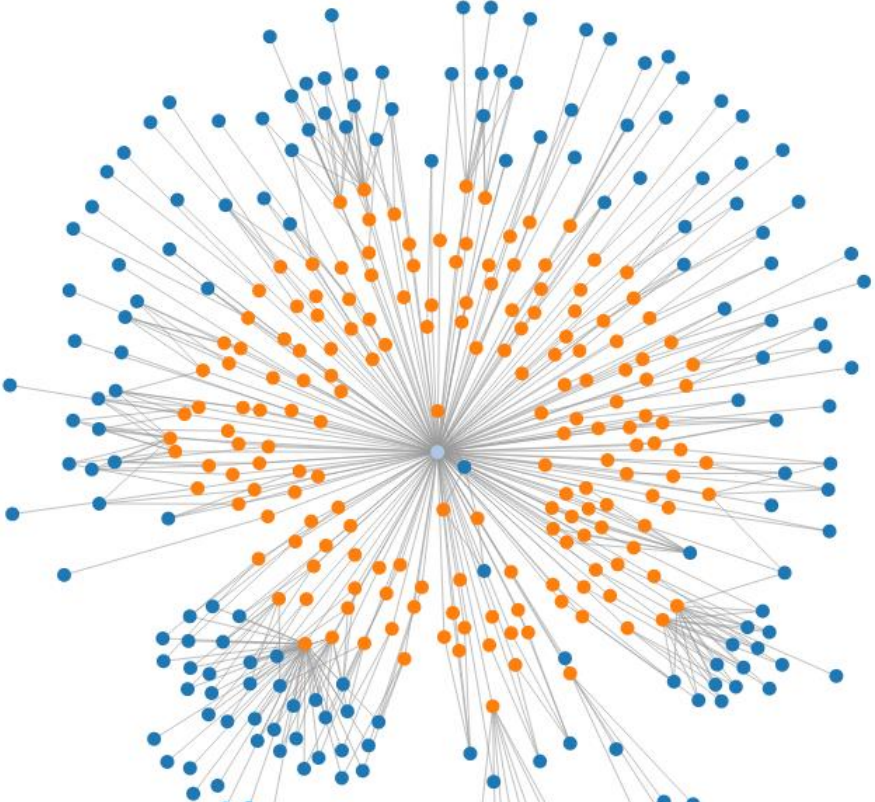
3.2.3.2 Obtener Listado de OERs.

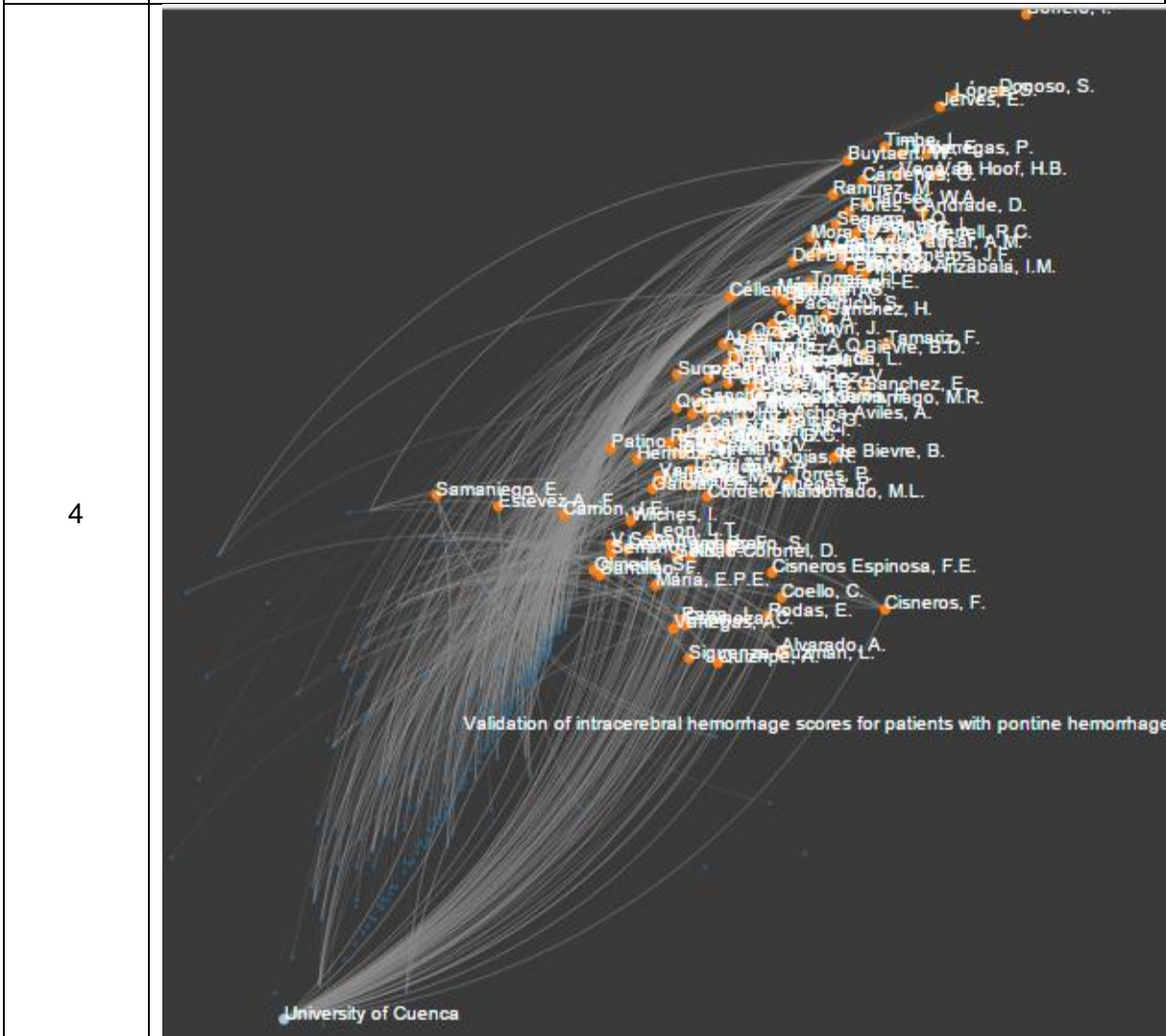
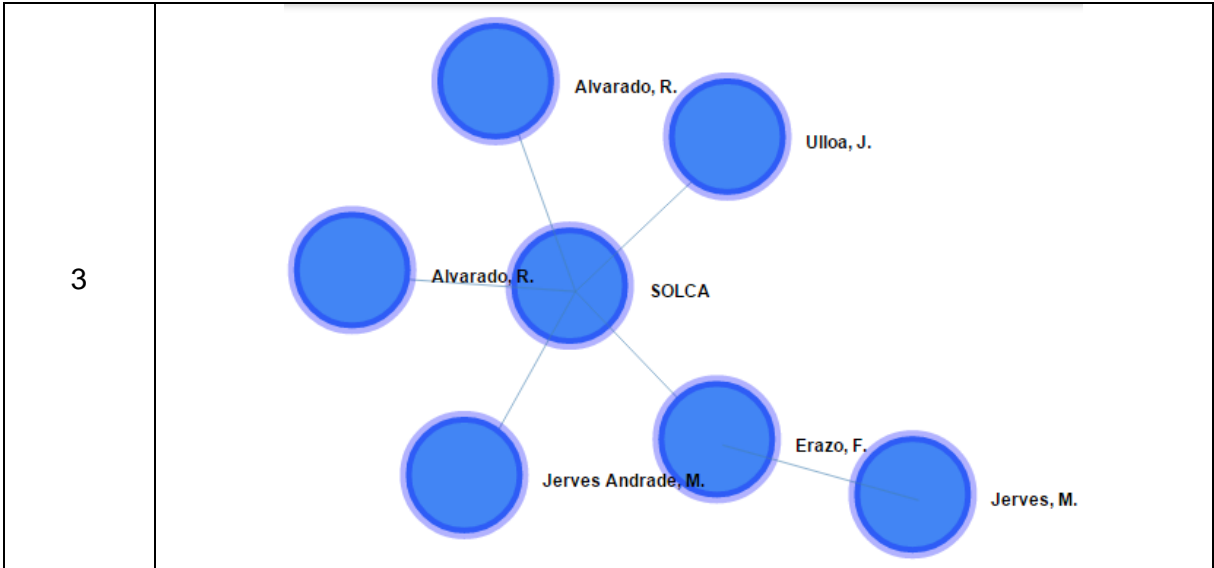
Obtiene un listado de los recursos educativos abiertos dependiendo de la institución y el tipo de recurso que se haya seleccionado previamente adicionalmente en este servicio se envía como parámetro e tipo de gráfica para la cual se requiere los datos a generar, como ya se sabe el resultado o respuesta está en formato JSON, peros su estructura cambia dependiendo de la gráfica seleccionada.

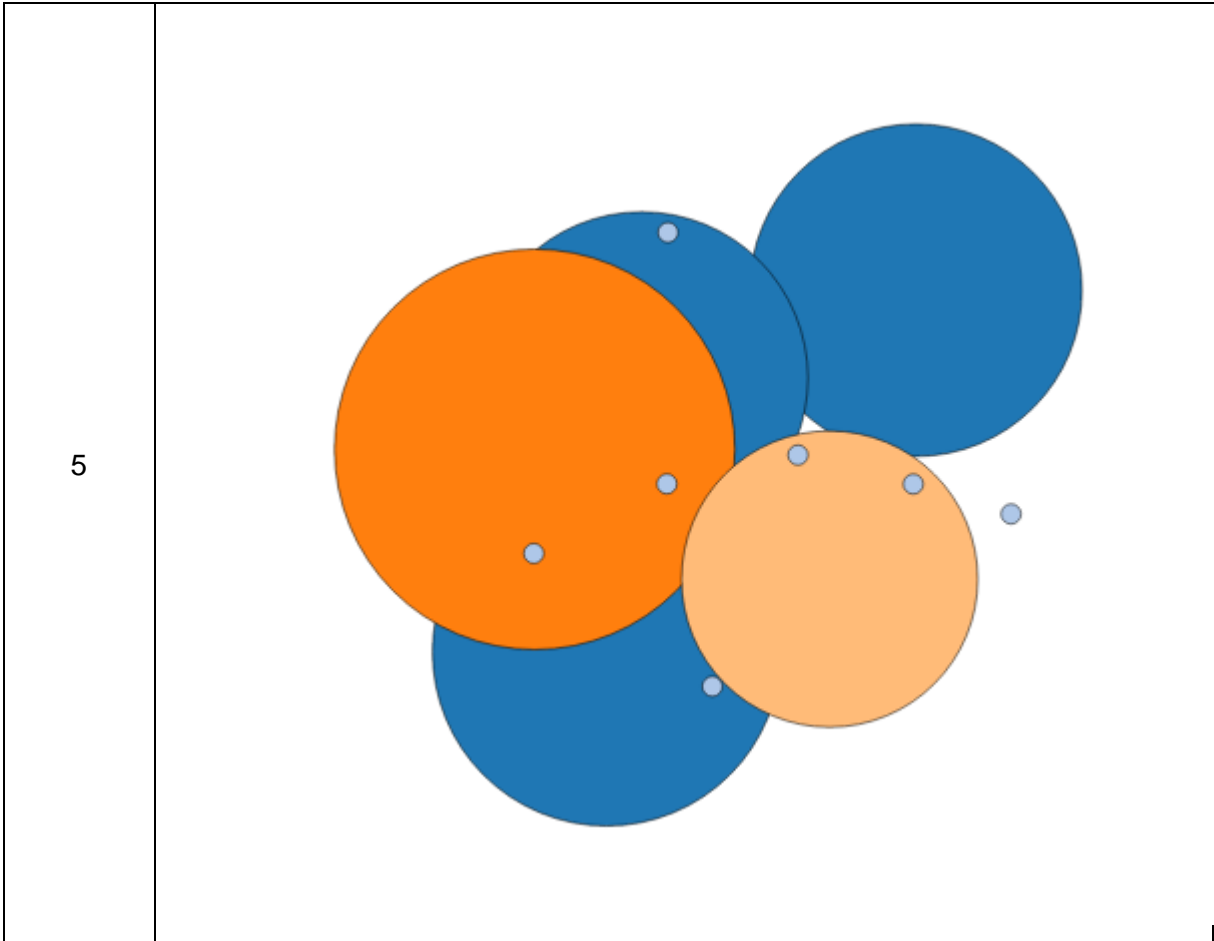
El tipo de grafica para hacer la consulta en el servicio está representado por un identificador numérico, es decir que como parámetro en la ruta de consulta iría su código que lo representa.

A continuación en la Tabla 20. Código de Grafica y grafica a la cual representan se detalla los códigos existentes actualmente y la gráfica a la cual representan

Tabla 20. Código de Grafica y grafica a la cual representan

Código	Gráfica
1	





Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

El detalle de este servicio se lo encuentra en la Tabla 21. Obtener Listado de OERs.

Tabla 21. Obtener Listado de OERs.

Descripción	Obtención de OERs	
URL de Solicitud	.../VDEOers-1.0-SNAPSHOT/webresources/parser/params	
Tipo de Petición:	GET	
Formato de Respuesta	JSON	
Parámetros	graph (Obligatorio)	Representa el tipo de grafico a presentar, es un parámetro numérico y cada valor que contiene identifica a cada una de las gráficas disponibles para la visualización.

		Ejemplo: 1
	publisher (Obligatorio)	Corresponde a la Institución en la cual se pretende realizar la búsqueda su valor corresponde a la URI que describa a la institución dentro del repositorio Serendipity OAR. Ejemplo: http://es.dbpedia.org/page/Universidad_T%C3%A9cnica_Particular_de_Loja
	type (Obligatorio)	Identifica a un Tipo de Recurso Educativo Abierto disponible en el repositorio Serendipity OAR, entre los principales se puede mencionar a Tesis o Maestrías. Su Ejemplo: http://purl.org/ontology/bibo/ThesisDegree
	search (Obligatorio)	Parámetro que se usa para filtrar datos en función a una cadena de texto ingresada manualmente, esta cadena de texto no es sensitiva a mayúsculas pueden ir en con mayúsculas o en minúsculas. Ejemplo: Cabrera

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Ejemplo:

GET: .../VDEOers-1.0-SNAPSHOT/webresources/parser/params?graph=3&publisher=http://dbpedia.org/resource/Universidad_Técnica_Particular_de_Loja&type=http://purl.org/ontology/bibo/Thesis&search=Carrion

Resultado:

Si se selecciona una gráfica tipo 3 el resultado devuelto seria como el que se muestra en la Figura 32. Listado de publicaciones generado para la gráfica tipo 3.

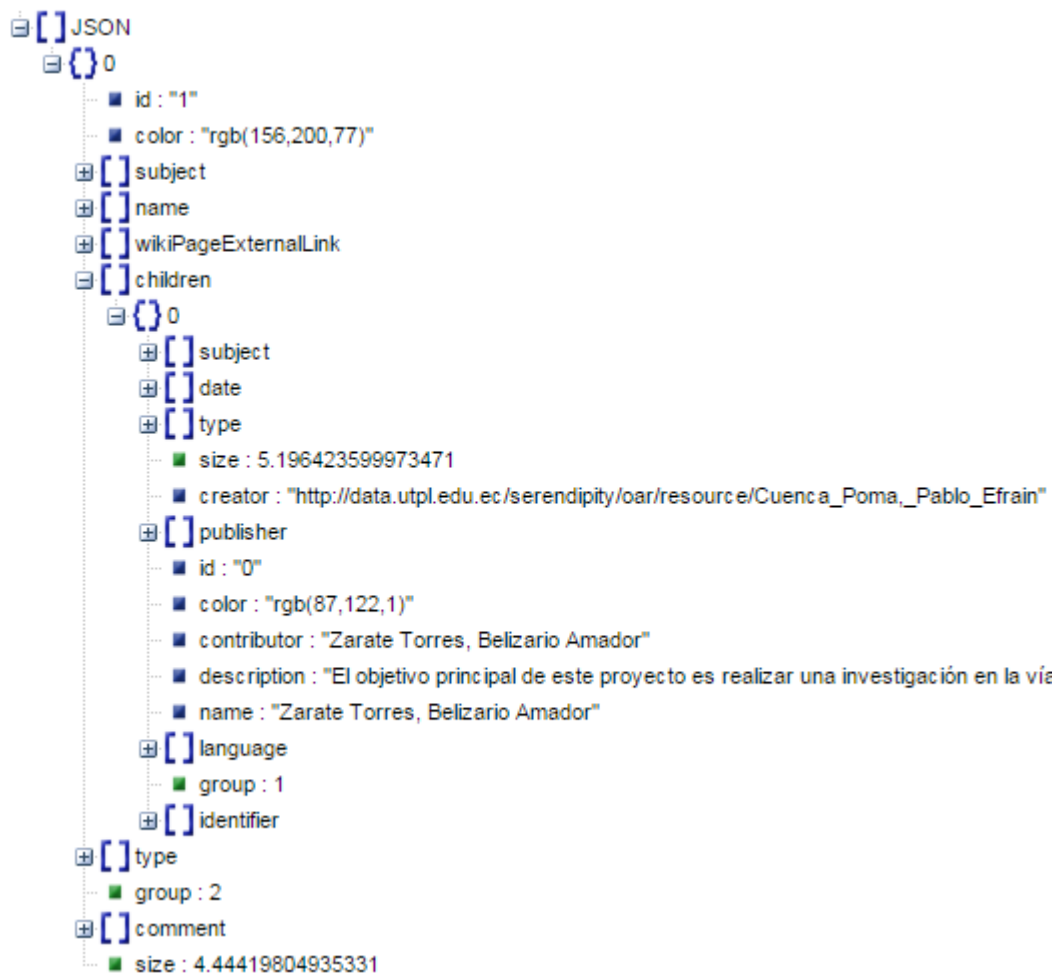


Figura 32. Listado de publicaciones generado para la gráfica tipo 3.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.3.3 Obtener Número de Oers por Tipo de Recurso, Parser.

Permite obtener un listado con el detalle de todos los tipos de recursos educativos abiertos que existen registrados en el EndPoint de Virtuoso juntamente con el número de recursos registrados por cada uno de ellos.

El resultado es un JSON con los datos de cada tipo de recurso y el número de recurso registradas por cada uno de ellos. El detalle de este servicio se lo encuentra en la Tabla 22. Descripción Obtención del Número de Oers por Tipo de Recurso.

Tabla 22. Descripción Obtención del Número de Oers por Tipo de Recurso.

Descripción	Obtener Número de Publicaciones por País.
URL de Solicitud	.../VDEOers-1.0-SNAPSHOT/webresources/parser/resourceType/{graph}
Tipo de Petición:	GET

Formato de Respuesta	JSON
Parámetros	--

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

Ejemplo:

.../VDEOers-1.0-SNAPSHOT/webresources/parser/resourceType/3

Resultado: Figura 33. JSON de Obtención del Número de Oers por Tipo de Recurso, Parser.

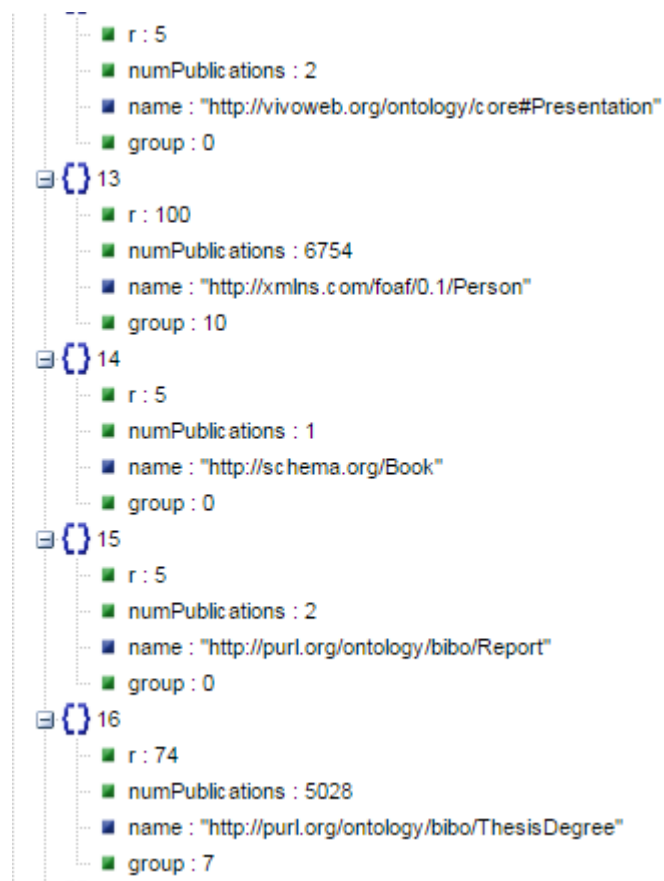


Figura 33. JSON de Obtención del Número de Oers por Tipo de Recurso, Parser.

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.4 Cliente.

El Cliente es el frontal de la aplicación, la parte que el usuario ve y manipula en donde se muestra las gráficas y datos que el usuario desea ver.

A través de elementos visuales se ingresan los valores necesarios para enviar a consultar al servicio web Parser este devuelve los resultados en formato JSON, una estructura de

acuerdo a la gráfica seleccionada según la Tabla 20. Código de Grafica y grafica a la cual representan.

En la Tabla 23. Especificación de Caso de Uso, Cliente se describe algunas de los elementos que se siguieron para el desarrollo del Cliente.

Tabla 23. Especificación de Caso de Uso, Cliente

Especificación del caso de uso: Cliente	
Código	3
Nombre	Cliente
Descripción	Este caso de uso describe el proceso que se siguió para realizar el cliente web de visualización
Actores	Cliente, WS Parser.
Precondición	El EndPoint así como los servicios web Consulta y Parse deben de estar disponibles y operativos para el correcto funcionamiento de la aplicación.
Post condición	Los parámetros de entrada deben de ser datos válidos y correctamente escritos para evitar problemas de consulta y mapeo de datos.
Flujo normal	<ol style="list-style-type: none"> 1. Abrir aplicación en la dirección http://carbono.utpl.edu.ec:8080/VDEOers-1.0-SNAPSHOT/ 2. Elegir un tipo de dirección entre los cuales se tiene Force Direct, Tree, Reddit, Resources Type. 3. Ingresar parámetros que atuan como filtro de datos. 4. Cargar Visualización
Excepciones	Si el usuario no ingresa los parámetros correctamente, no se realizara ninguna acción por falta de datos.
Anotaciones	Ninguna

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

El acceso a la aplicación está disponible en la dirección, <http://carbono.utpl.edu.ec:8080/VDEOers-1.0-SNAPSHOT> y en ella se puede escoger el tipo

de grafico que se desee visualizar Force Direct, Tree, Reddit, Resources Type. Según el tipo de grafico seleccionado se puede ver los datos adicionales y filtros de información.

Para mayores referencias ver Anexo 3: MANUAL DE APLICACIÓN.

3.2.5 Preguntas que Resuelven Estas Aproximaciones.

A continuación se presentan ciertas interrogantes las cuales resuelven mediante la visualización de datos.

De acuerdo al filtro de datos:

3.2.5.1 ¿Cuáles son los recursos para una determinada Institución?

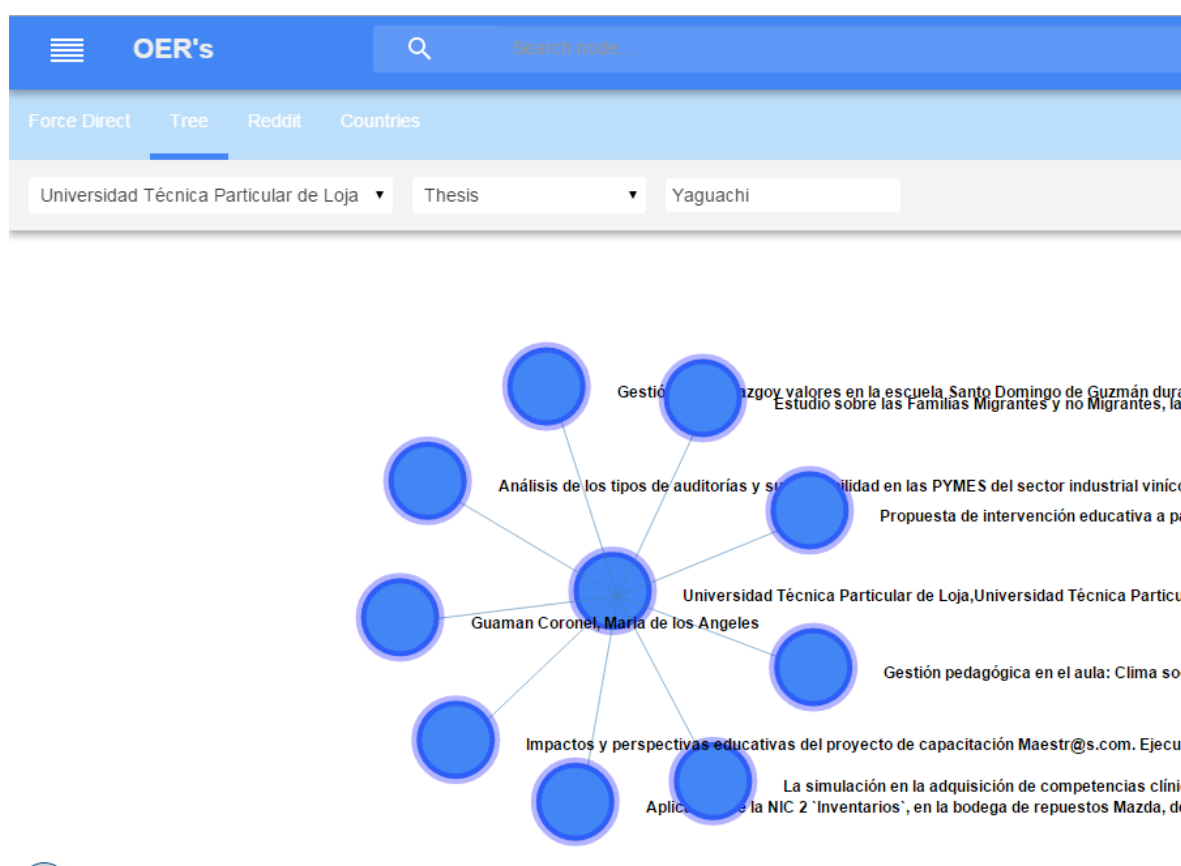


Figura 34. Autores para una determinada Institución.

Fuente: Servicios Web VDE

Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.5.2 ¿Cuáles son las publicaciones que han realizado un los Autores de un determinada Institución?

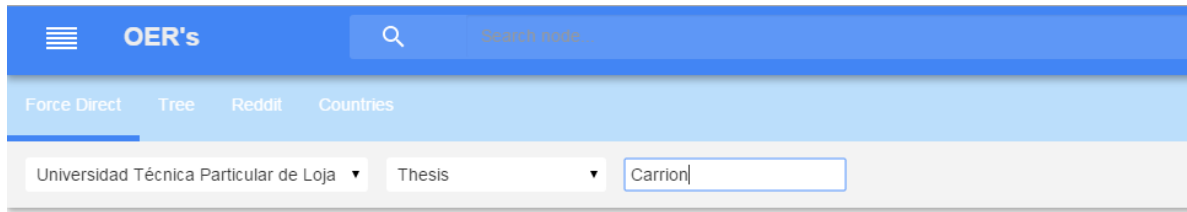


Figura 35. Publicaciones que han realizado un los Autores de un determinada Institución
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.5.3 ¿Cuáles son los recursos que ha realizado un determinado autor?

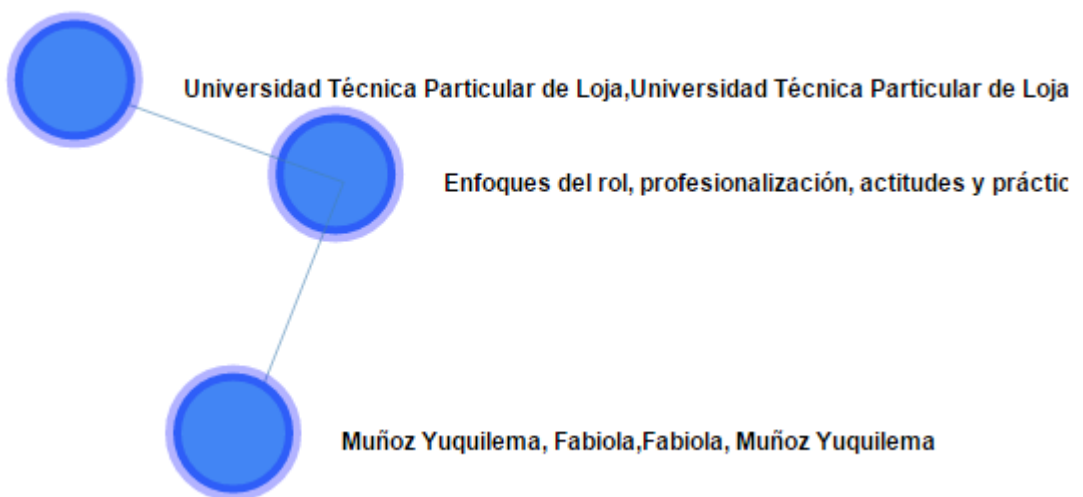


Figura 36. Recursos que ha realizado un determinado Autor.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

3.2.5.4 ¿Cuáles son el número de publicaciones que ha realizado cada Tipo de Recursos?

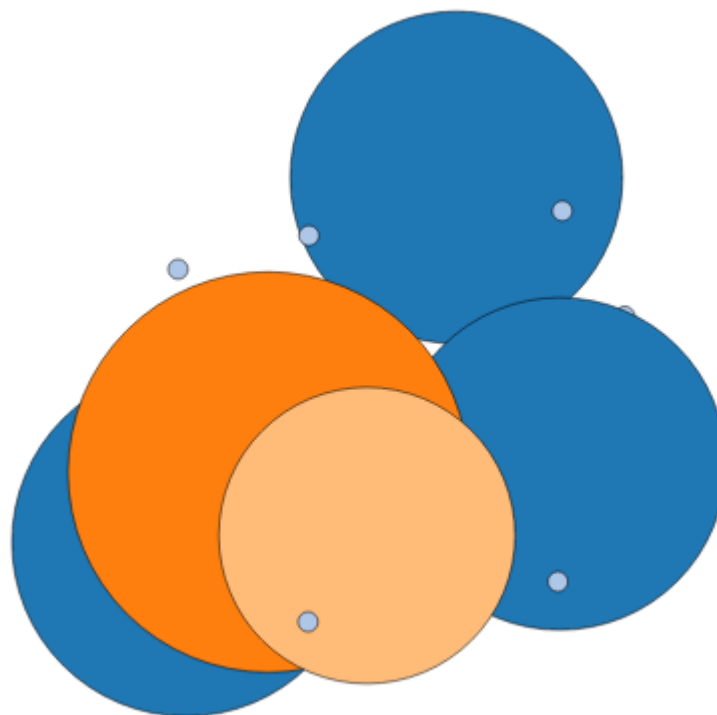


Figura 37. Número de recursos por Tipo de Recurso.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

DISCUSIÓN

El presente trabajo de investigación va ligado con la construcción de la aplicación de visualización de datos enlazados que ha permitido el uso de varias tecnologías (Sparql, Java, REST, HTML, JavaScript, CSS), el uso de cada una de ellas ha ayudado en gran medida para armar esta aplicación permitiendo así mismo la integración de servicios REST frente a datos públicos almacenados en un TripleStore como lo es Virtuoso.

Descripción de Datos

El proceso de análisis de datos almacenados en un TripleStore involucra implícitamente manejar conceptos de triplas, sujeto, predicado y objeto que es el la base del RDF además de otros conceptos de datos enlazados como la Grafos, Vocabularios, Literales, Clases, Propiedades, etc.

El análisis de uno o varios vocabularios que describa y enlace a cada uno de los elementos del predicado disponibles para los recursos educativos abiertos es una tarea fundamental, aprovechando los principios de linked data de reutilizar y redistribuir los contenidos existentes en la web, se utilizó el buscador LOV¹² (Linked Open Vocabularies) para describir los vocabularios que acorde a los datos obtenidos, dándole a estos datos un significado y una identidad dentro de un segmento determinado.

Patrones de Visualización

La visualización de datos es una herramienta muy útil a la hora de analizar e interpretar muchos datos ya que facilitan el rápido entendimiento y comprensión de los que se observa, los patrones de visualización ayudan a la estructuración de los mismos para representarlos en una determinada herramienta de visualización.

En el proceso de definición de patrones de visualización lo primero que se realizó fue una búsqueda de las herramientas disponibles que cumplen con esta tarea, en la red se pueden observar muchas de ellas, de preferencia se escogió las que trabajan en la web haciendo usos de tecnologías como HTML, CSS y JavaScript y con el intercambio de datos en formato JSON, es así como por ser estas tecnologías abiertas y universalmente aceptadas por la mayoría de las personas; se encontraron en la red un conjunto de herramientas de las cuales se evaluaron y probaron algunas de ellas para determinar con cuales se trabajara.

Durante la evaluación de la herramientas de visualización se revisó principalmente estos servicios Exhibit, Arbor, D3 e InfoVis algunos de ellos cuentan solamente con un tipo de grafica como es el caso de Exhibit y Arbor y otras que cuentan una variedad de ejemplos de

¹² <http://lov.okfn.org/dataset/lov/>

los cuales se puede elegir y validar algunos de ellos y estudiar su estructura, estos son D3 e InfoVis.

Estos servicios trabajan con datos en formato JSON, analizar su estructura es una tarea importante ya que al hacerlo da un entendimiento más claro de lo que se quiere modelar, además de ir identificando los elementos necesarios con los cuales trabajan y los elementos que son opcionales en la estructura JSON, otra de las ventajas que brinda el análisis de la estructura JSON es identificar una estructura similar o igual a otra que ya se hay validado dando esta opción la posibilidad de que con un mismo patrón de visualización graficar dos o más graficas con la misma estructura, puede ser otras graficas del mismo servicio como de diferentes servicios siempre y cuando la estructura sea la parecida.

Servicios Web

Como se lo menciono en el capítulo

CAPÍTULO III: SERVICIOS WEB uno de los objetivos y necesidades de la aplicación es la modularidad de la aplicación, es decir que sus elementos sean intercambiables o reemplazables, así por ejemplo si se desea trabajar con otros datos que el esfuerzo de este cambio solo involucre en cambiar los datos y que el resto de la aplicación continúe trabajando normal. Una de las formas de enfrentar esa situación es mediante la construcción de servicios web asignando a cada uno de ellos una tarea específica a realizar.

Servicio de Consulta.

Con los datos listos en el EndPoint de virtuoso y los patrones de visualización definidos lo que se requiere es consultar estos datos desde virtuoso hacia la aplicación tal cual se lo

realiza con base de datos relacionales, gracias a la ayuda de las librerías como jena y virtuoso para java se puede decir que la operación de consulta de los datos desde un TripleStore se puede realizar de forma efectiva y sencilla siempre teniendo en mente que se trabaja en con datos RDF y que las consultas son realizadas mediante SPARQL. El servicio web debe dar una respuesta cuando se haga una petición sobre él, esta respuesta será un JSON estructurado en un formato estándar, así que es necesario contar con una estructura de clases java requeridas para serializar los datos consultados en un JSON con formato definido.

Servicio de Parser.

El resultado de la consulta de datos tiene que mapearse a un formato específico que corresponde a la estructura de datos de una gráfica seleccionada.

El servicio de Parser se encarga de integrar los datos de la consulta (JSON Base) con la estructura de datos necesaria para una gráfica específica, en este punto es donde los patrones de visualización se usan ya que dependiendo de la petición del cliente se llamara a una u otra gráfica, para satisfacer esta demanda el servicio de Parser, necesita serializar los datos consultados con la estructura de datos requerido por el patrón de visualización para ello se ha determinado a nivel de servicio la definición de clases requerida para estructuración de datos de cada patrón de visualización requerido.

El nivel de complejidad en el modelado de las clases depende de la complejidad de estructura JSON que requiera el patrón de visualiza sobre todo resulta complicado modelar estructuras JSON que anidan estructuras del mismo tipo de objetos aquellas pueden llegar a tener N subniveles de datos haciendo del proceso de serialización un tanto complicado.

Cliente Web.

El cliente es la parte visible de la aplicación, la parte en la que el cliente o usuario final interactúa con la aplicación, realiza las peticiones de consulta a los servicios y presenta los resultados.

La consulta de datos a visualizar se lo realiza mediante filtro de datos para que no solo se pueda tener resultados más precisos a la hora de interpretar los datos sino que también evitar problemas de sobrecarga de datos al navegador y termine por cerrarse el navegador inesperadamente, se considera que no se debe arruinar la experiencia del usuario por querer satisfacer algunas necesidades funcionales.

La consulta y visualización de datos filtrados por parámetros de consulta y representados en el navegador demuestra que la visualización de datos enlazados extraídos desde un TripleStore se puede realizar de forma transparente usando tecnologías basadas en la Web.

Para el correcto funcionamiento de la aplicación así como también aportar a la modularidad y mapeo entre servicios web se hizo uso de archivos JSON de configuración definidos previamente para brindar soporte ya hacer cambios a las estructuras JSON utilizadas en las diferentes etapas de la aplicación si es requerido, esto con el objetivo de no tener que depender demasiado del código fuente de la aplicación sino de elementos externos que faciliten la configuración.

CONCLUSIONES

Al terminar el presente trabajo investigativo se puede concluir lo siguiente:

- El análisis de datos almacenados en el EndPoint Serendipity OAR es una tarea fundamental ya que permite tener una idea clara de la estructura y objetos con los que se va trabajar así como el uso de tripletas RDF implica tener en mente otro paradigma de almacén de datos, en donde el concepto de Grafo RDF y tripleta es una parte importante para este efecto. De la misma forma la reutilización de predicados es una de las actividades importantes ya que con el uso de uno de ellos se identifica y se da sentido a la relación sujeto, objeto; de no ser así se tendrá una relación ambigua y no se estaría cumpliendo con el objetivo de Linked Data de facilitar la relaciones y el entrelazado de datos para que sea mejor aprovechada por los usuarios y motores de búsqueda.
- Definir patrones de visualización garantiza que de una misma petición de consulta se pueda generar la estructura de datos necesaria para graficar varias tipo de gráficas esto depende del número de patrones de visualización definidos en el servicio web Parser.
- El proceso de extracción de datos desde el EndPoint Serendipity OAR y realizar operaciones sobre ellos o exponerlos bajo demanda a través servicios web realizado en el servicio web de Consulta implica el uso de un JSON estándar que sea el mismo para todas las peticiones de consulta y que en él se encuentre la relación entre cada uno de sus nodos, a partir de aquí poder definir nuevas estructura JSON para las diferentes gráficas requeridas. Además que gracias a las herramientas y librerías disponibles hoy en día esto es una tarea que se puede realizar de forma efectiva y sencilla
- El uso de varios patrones de visualización para varias graficas hace necesario la creación de un servicio que gestione las peticiones del usuario y valide cual es la gráfica seleccionada (Parser), en esta etapa es muy importante el correcto mapeo entre etiquetas JSON y el uso de archivos JSON que funcionan como tabla equivalencias permite realizarlo de forma dinámica y precisa, debido a que cada patrón varia uno del otro, con esto se evita errores en la visualización de la gráfica.
- Diseñar el cliente web implica estructurar la información que visualizara el usuario y organizarla de forma correcta para que esta tome sentido cuando se muestre, además de usar los compontes necesarios para realizar el filtro de datos y búsqueda

de elementos específicos, en este sentido las herramientas y tecnologías basadas en la web de hoy en día permiten la creación de graficas con una buena calidad de diseño y animaciones lo cual es de gran ayuda para capturar la atención del usuario así como también mejorar su experiencia de uso.

RECOMENDACIONES

Durante el proceso de elaboración del presente trabajo investigativo se ha extraído las siguientes recomendaciones:

- El análisis de datos que proceden de diferentes fuentes de información en tripletas semánticas requiere del uso de vocabularios semánticos para la asociación de recursos, por lo cual se debe procurar utilizar vocabularios existentes y compartidos en la web; en el caso de requerir la creación de uno, seguir los estándares internacionales para la creación y publicación de vocabularios, esto para garantizar la interoperabilidad de recursos y aplicaciones web.
- Trabajar con patrones de visualización ha ayudado en la visualización de graficas con mismo formato de petición, incluir un nuevo patrón requiere que se realice su definición y mapeo desde el JSON estándar hacia la nueva estructura que se desee incluir.
- El uso de archivos JSON que se utilizan como equivalencias en la diferentes etapas de la aplicación permite para el correcto mapeo entre JSONs, si se quiere cambiar alguna etiqueta en alguno de los JSONs generados en estos archivos hay que realizarlo tomando como referencia el valor que se tiene en la etiqueta actual remplazar con el valor que se quiere remplazar
- El uso de herramientas y tecnologías basadas en la web para realizar trabajos y actividades de visualización de datos debido a que el desarrollo web está en auge y cada vez hay una comunidad más grande que de una u otra forma da soporte para que estas tecnologías vayan evolucionando con el tiempo, no requiere una curva de aprendizaje compleja y cuenta con un sin número de recursos, tutoriales y documentación para su aprendizaje.
- A la hora de consultar grandes cantidades de datos en una aplicación web es mejor obtener esta información según sea como lo requiera el usuario y no traer toda la información al inicio y filtrar en memoria, esto puede ocasionar que el navegador trabaje de forma lenta o deje de responder por sobrecarga de información.

BIBLIOGRAFÍA

- Amat, C. B. (2009). La visualización de resultados de búsqueda y los gestores bibliográficos personales: una nota técnica sobre RefViz 2. *El profesional de la información*, 18(1), 87-91.
- Barrueco, J. M., & Subirats Coll, I. (2003). OAI-PMH: protocolo para la transmisión de contenidos en internet. *El profesional de la información*, 99-106.
- Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data -- The story so far. *International Journal on Semantic Web and Information Systems*, 26.
- CENTRE FOR EDUCATIONAL RESEARCH AND INNOVATION. (2007). *THE EMERGENCE OF OPEN EDUCATIONAL RESOURCES*. OECD.
- Cooley, M. (1987). *Architect or Bee?* London: UK.
- Demchenko, Y., Paola, G., de Laat, C., & Membrey, P. (2013). Addressing Big Data Issues in Scientific Data Infrastructure. *IEEE*, 48 - 55.
- ELSEVIER. (2014). *ELSEVIER*. Recuperado el 9 de Julio de 2014, de <http://www.elsevier.com/online-tools/scopus/features>
- Erling, O. (s.f.). *OpenLink Software*. Recuperado el 23 de Julio de 2014, de <http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSRDFWP>
- Fensel, D., Bussler, C., Ding, Y., & Omelayenko, B. (2005). The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 19.
- Few, S. (2013). Data Visualization for Human Perception. *The Encyclopedia of Human-Computer Interaction*, II(http://www.interaction-design.org/encyclopedia/data_visualization_for_human_perception.html).
- Friendly, M. (2009). Milestones in the history of thematic cartography, statistical graphics, and data visualization. *datavis.ca*, 1(1), 2.
- González Díez, A. (13 de Junio de 2013). *Outliers Collective*. Recuperado el 23 de Julio de 2014, de http://assets.outliers.es/cursos/viz_madrid/Visualizacion%20de%20Datos%20con%20D3.js%20Temp.pdf
- Gracy, K. F. (2014). Archival description and linked data: a preliminary study of opportunities and implementation challenges. *Springer Science + Business Media Dordrecht 2014*, 1-6.
- Graphite; ARC2. (s.f.). *RDF Triple-Checker*. Recuperado el 05 de Diciembre de 2015, de <http://graphite.ecs.soton.ac.uk/checker/>
- Gustavo, A. (2008). SOAP. *Information Communication System Research Group*, I(1), 45.
- JSON. (09 de Marzo de 2010). *Introducing JSON*. Recuperado el 22 de Febrero de 2014, de <http://json.org/>

- Keller, T., & Tergan, S.-O. (2005). *Visualizing Knowledge and Information: An Introduction*. Springer, 1-23.
- Lamarca Lapuente, M. J. (08 de Diciembre de 2013). *Hipertexto*. Recuperado el 11 de Junio de 2014, de El nuevo concepto de documento en la cultura de la imagen: <http://www.hipertexto.info/documentos/rdf.htm>
- LODE. (21 de Mayo de 2014). *SCoRO, the Scholarly Contributions and Roles Ontology*. Recuperado el 25 de Junio de 2014, de <http://www.essepuntato.it/lode/http://purl.org/spar/scoro>
- McCandless, D. (Julio de 2010). The beauty of data visualization. EEUU.
- Meyer, R. (2009). Knowledge Visualization. *Media Informatics Advanced Seminar on Information Visualization*, 8.
- OpenLink Softwar. (2009). *OpenLink Softwar*. Recuperado el 04 de Mayo de 2014, de OpenLink Virtuoso Open-Source Edition: Downloads: <http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSDownload#Jena%20Provider>
- Sagiroglu, S., & Sinanc, D. (2013). Big Data: A Review. *IEEE*, 42 - 47.
- Saquicela, V., Vilches-Blázquez, L. M., & Corcho, O. (2012). Adding Semantic Annotations into (Geospatial) RESTful Services. *International Journal on Semantic Web and Information Systems*, 8(2), 23.
- Schema.org. (21 de Junio de 2011). *schema.org*. Recuperado el 02 de Junio de 2014, de Getting started with schema.org: <http://schema.org/docs/gs.html>
- W3C. (11 de 02 de 2004). Recuperado el 19 de 03 de 2014, de Web Services Architecture: <http://www.w3.org/TR/ws-arch/>
- W3C. (10 de Febrero de 2004). W3C. Recuperado el 23 de Julio de 2014, de Resource Description Framework (RDF): Concepts and Abstract Syntax: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- W3C. (20 de Noviembre de 2011). W3C. Recuperado el 22 de Febrero de 2014, de Guía Breve de Servicios Web: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
- W3C. (27 de Septiembre de 2012). *R2RML: RDB to RDF Mapping Language*. Recuperado el 25 de Abril de 2014, de <http://www.w3.org/TR/r2rml/>
- Weibel, S. (1997). The Dublin Core: A Simple Content Description Model For Electronic Resources. *Bulletin of the American Society for Information Science*, 9-11.
- Wikipedia. (12 de Febrero de 2010). *Wikipedia*. Recuperado el 28 de Abril de 2014, de Sabiduría: <http://es.wikipedia.org/wiki/Sabidur%C3%ADa>
- Wikipedia. (31 de Diciembre de 2013). *Wikipedia*. Recuperado el 22 de Febrero de 2014, de Recursos educativos abiertos: http://es.wikipedia.org/wiki/Recursos_educativos_abiertos#cite_note-Johnstone-1

Wikipedia. (29 de Noviembre de 2013). *Wikipedia*. Recuperado el 25 de Junio de 2014, de
Dublin Core: http://es.wikipedia.org/wiki/Dublin_Core

Wikipedia. (19 de Mayo de 2014). *Wikipedia*. Recuperado el 11 de Junio de 2014, de
Scopus: <http://es.wikipedia.org/wiki/Scopus>

Wikipedia. (11 de Marzo de 2014). *Wikipedia*. Recuperado el 19 de Marzo de 2014, de Web
Services Description Language:
http://en.wikipedia.org/wiki/Web_Services_Description_Language

Wikipedia. (11 de Marzo de 2014). *Wikipedia*. Recuperado el 19 de Marzo de 2014, de
Simple Object Access Protocol (SOAP):
http://es.wikipedia.org/wiki/Simple_Object_Access_Protocol

ANEXOS

ANEXO 1. DIAGRAMA DE SECUENCIA APLICACION DE CONVERSION.

En el siguiente diagrama se pretende describir de una forma global el proceso por el cual se tomó como base para la creación de la aplicación que convierte los datos a se tenían inicialmente a para así convertirlos y exportarlos a virtuoso.

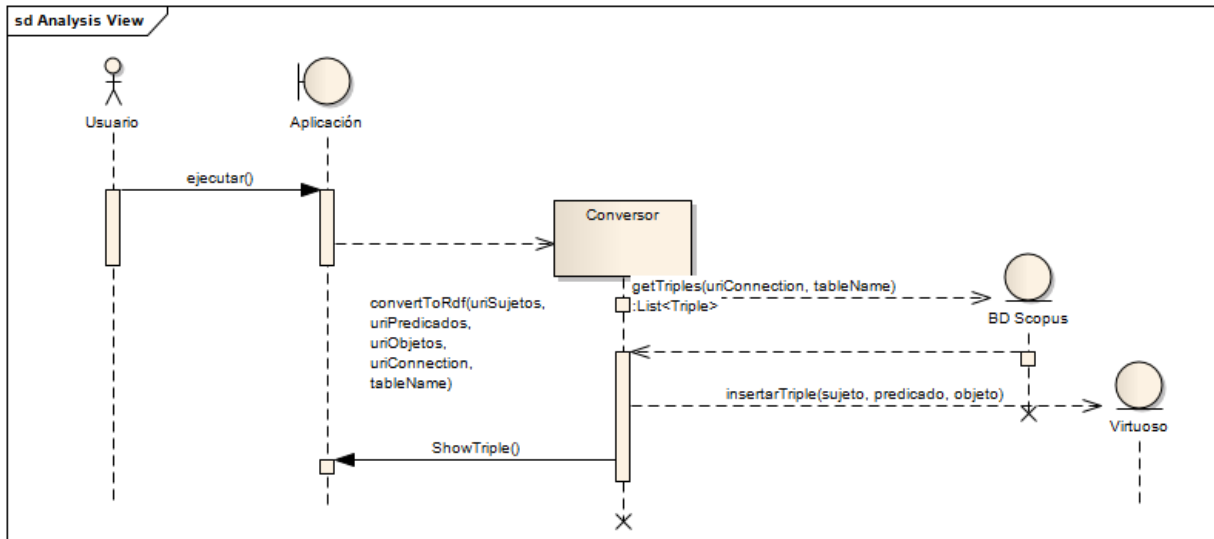
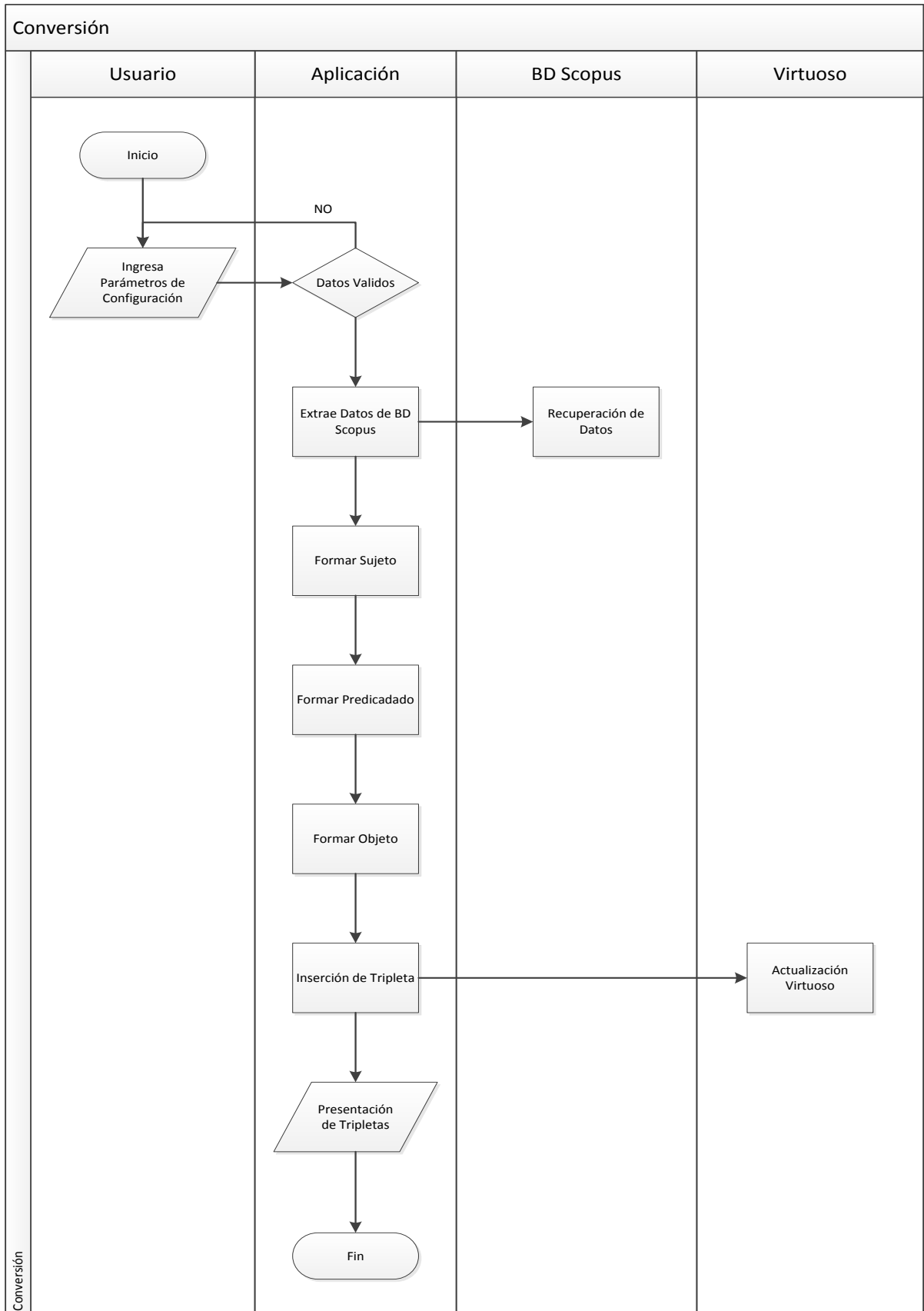


Figura 38. Diagrama de Secuencia – Conversor

Fuente: Datos Serendipity

Elaboración: Jonathan Yaguachi, Octubre 2015

ANEXO 2. PROCESO DE CONVERSIÓN



ANEXO 3. MANUAL DE APLICACIÓN

Introducción.

El presente documento pretende mostrar de forma funcional y secuencial las características de la aplicación de visualización de datos enlazados Scopus. Se cuenta con tres gráficas en las que se puede observar la información de tipo nodo - relación estos datos provienen de Servicios Web que vienen en formato JSON.

La información presentada corresponde a datos de referencias bibliográficas, resúmenes, artículos y citas extraídos desde Scopus, Estos datos han sido tratado y almacenados en una Base de Conocimientos como Virtuoso.

Descripción.

La aplicación está construida usando herramientas y tecnologías basadas en la web, es decir para su elaboración se hizo uso de tecnologías como HTML, CSS y Javascript; principalmente en su interfaz. La comunicación de la misma se la realizó a través de Servicios Web usando datos en formato JSON.

A continuación se presenta la vista inicial de la aplicación una vez se haya ejecutado.

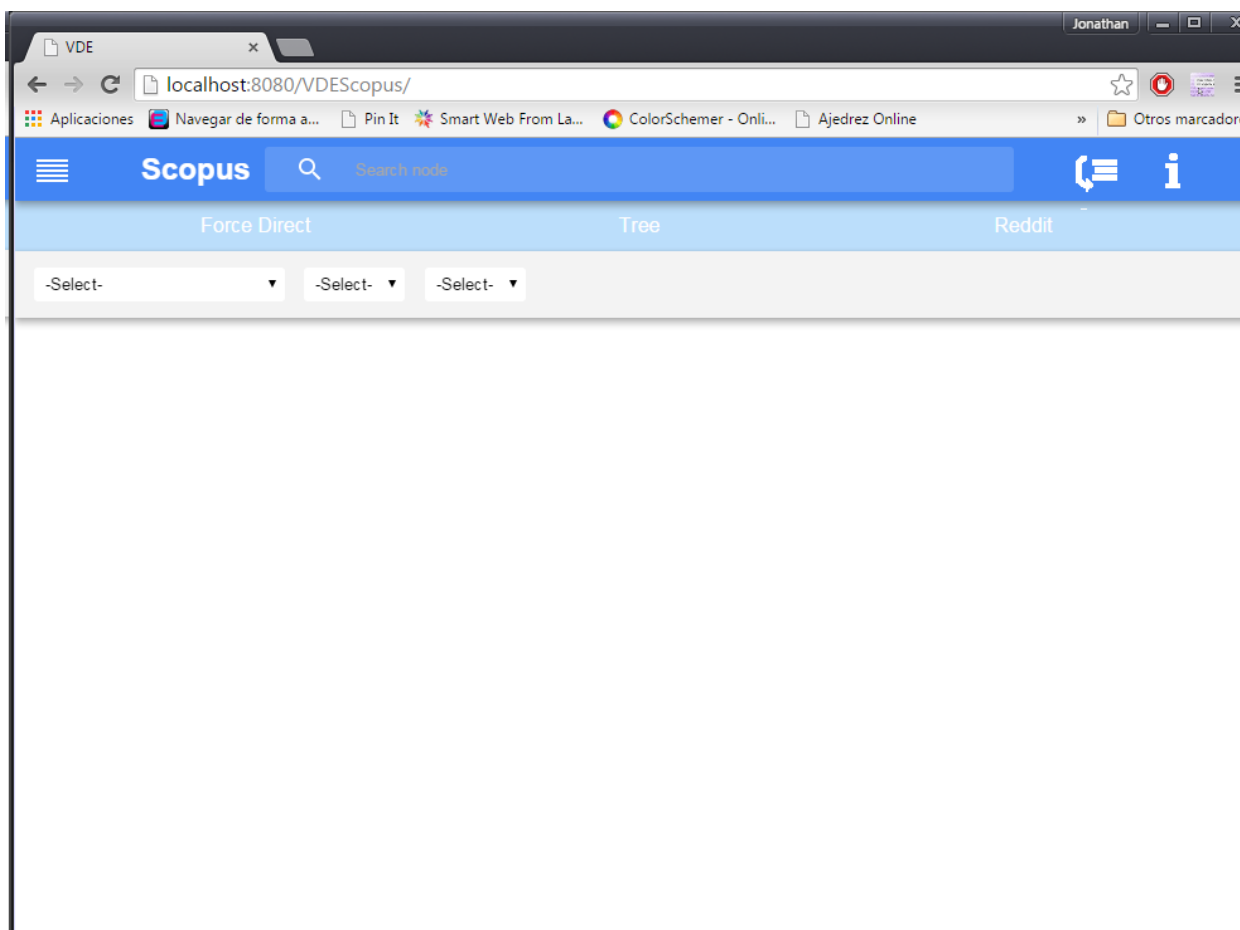


Figura 39. Frontal de la Aplicación.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Elementos Principales.

Menú.

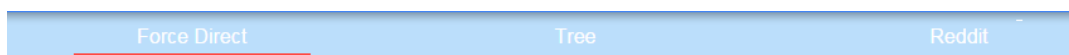


Figura 40. Menú de la Aplicación.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

En el menú se cuenta con básicamente tres opciones, cada uno de ellos representa una gráfica diferente a visualizar, se puede seleccionar cualquiera de ellas en el momento que se desee, por defecto carga la primera opción Force Direct.

Campos de Filtro de Datos.

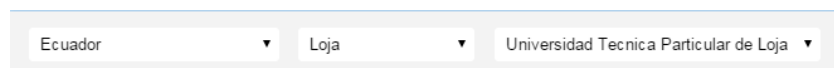


Figura 41. Campos Para el Filtro de Datos.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Scopus tiene un sin número de referencias bibliográficas de muchas instituciones en el mundo, es por esa razón que resulta complicado visualizar toda esta información en una solo vista es por eso que se crearon estos campos que ayudan a filtrar la información por tres elementos específicos como son el país, ciudad e institución.

Gráfica

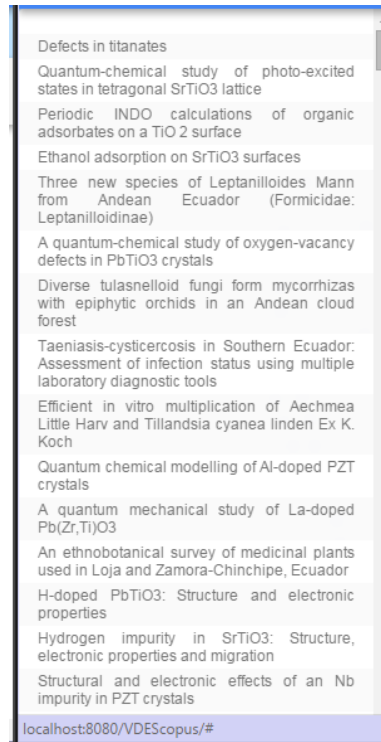


Figura 44. Lista de Nodos.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

Si se decide mostrar el listado de nodos desplegados para la gráfica actual, se mostrará un elemento en la parte lateral izquierda en donde se puede encontrar con el nombre de cada nodo, se puede hacer clic sobre el nombre y le mostrará su ubicación en la Gráfica.

Búsqueda de Nodos.

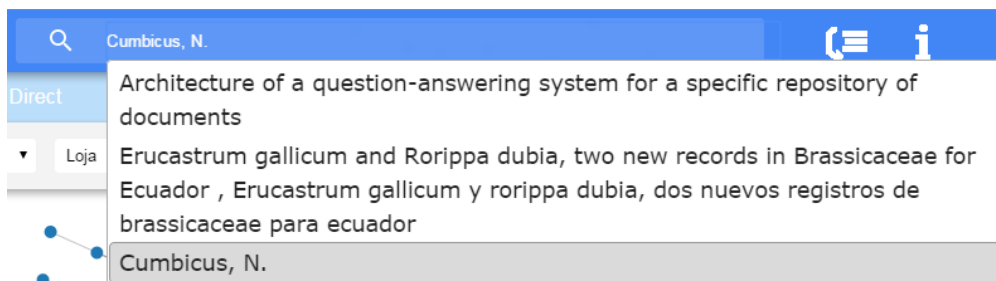


Figura 45. Búsqueda de Nodos
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

Campo ubicado en la parte superior de la aplicación el cual le permite buscar en cualquier momento que desee un nodo en la gráfica. La búsqueda de nodos consiste en escribir todo

o parte del nombre del nodo a buscar y automáticamente desplegará una lista de autocompletado que le ayudará a elegir una opción en caso de no recordar exactamente el nombre del nodo buscado.

Pop up, Información de Nodo.

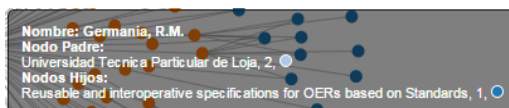


Figura 46. Pop up, Información de Nodo.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Un pop up es un elemento que no está visualmente en la aplicación y que se muestra tras realizar un evento, en este caso se muestra al pasar el cursor sobre un determinado nodo.

La información que contiene este pop up es información básica de cada nodo como lo es el nombre de nodo, el nodo padre, y un pequeño listado de los nodos hijos que le corresponden.

Información de Nodo.



Figura 47. Información de Nodo.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Al hacer clic en algún nodo en específico automáticamente se desplegará un elemento en la parte lateral derecha de la pantalla, en esta sección se presenta la información completa de cada nodo, si se decide hacer clic sobre otro nodo automáticamente se irá actualizando esta información.

Mostrar / Ocultar Información de Nodo.



Figura 48. Mostrar / Ocultar Información de Nodo.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Opción disponible en la parte superior derecha de la pantalla cuya función es básicamente mostrar u ocultar la información del nodo. Si ya no se desea ver la información del nodo se puede hacer clic sobre esta opción y la información del nodo desaparece dando una vista más clara de la gráfica.

Mostrar/Ocultar panel de Búsqueda Avanzada.



Figura 49. Mostrar/Ocultar panel de Búsqueda Avanzada.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Opción disponible en la parte superior derecha de la pantalla al lado izquierdo del botón Mostrar / Ocultar Información de Nodo. Presionando sobre esta opción se desplegará automáticamente un panel con campos y opciones que le ayudaran en su búsqueda.

Búsqueda Avanzada.

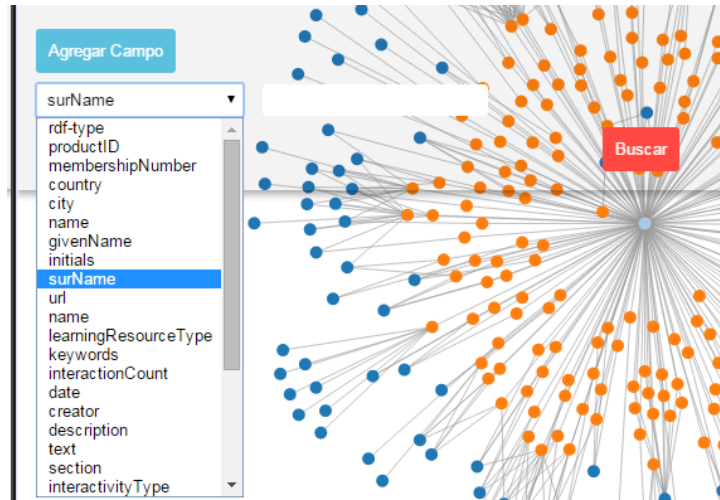


Figura 50. Búsqueda Avanzada.
 Fuente: Servicios Web VDE
 Elaboración: Jonathan Yaguachi, Octubre 2015

Panel que se despliega por debajo de los Campos de Filtro, es una serie de campos que le permitirán hacer una búsqueda por cada uno de los elementos de información de que contenga cada nodo. Consta de dos elementos básicamente, un combo para elegir el que elemento buscar y un campo donde se ingresa el valor a buscar. Si se desea más campos para complementar la búsqueda se presiona sobre el botón Agregar Campo y agregara un nuevo combo y campo de búsqueda en donde se puede agregar nuevos parámetros de búsqueda. Finalmente se procede a realizar la búsqueda presionando sobre el botón Buscar.

Gráficas

Force Direct.

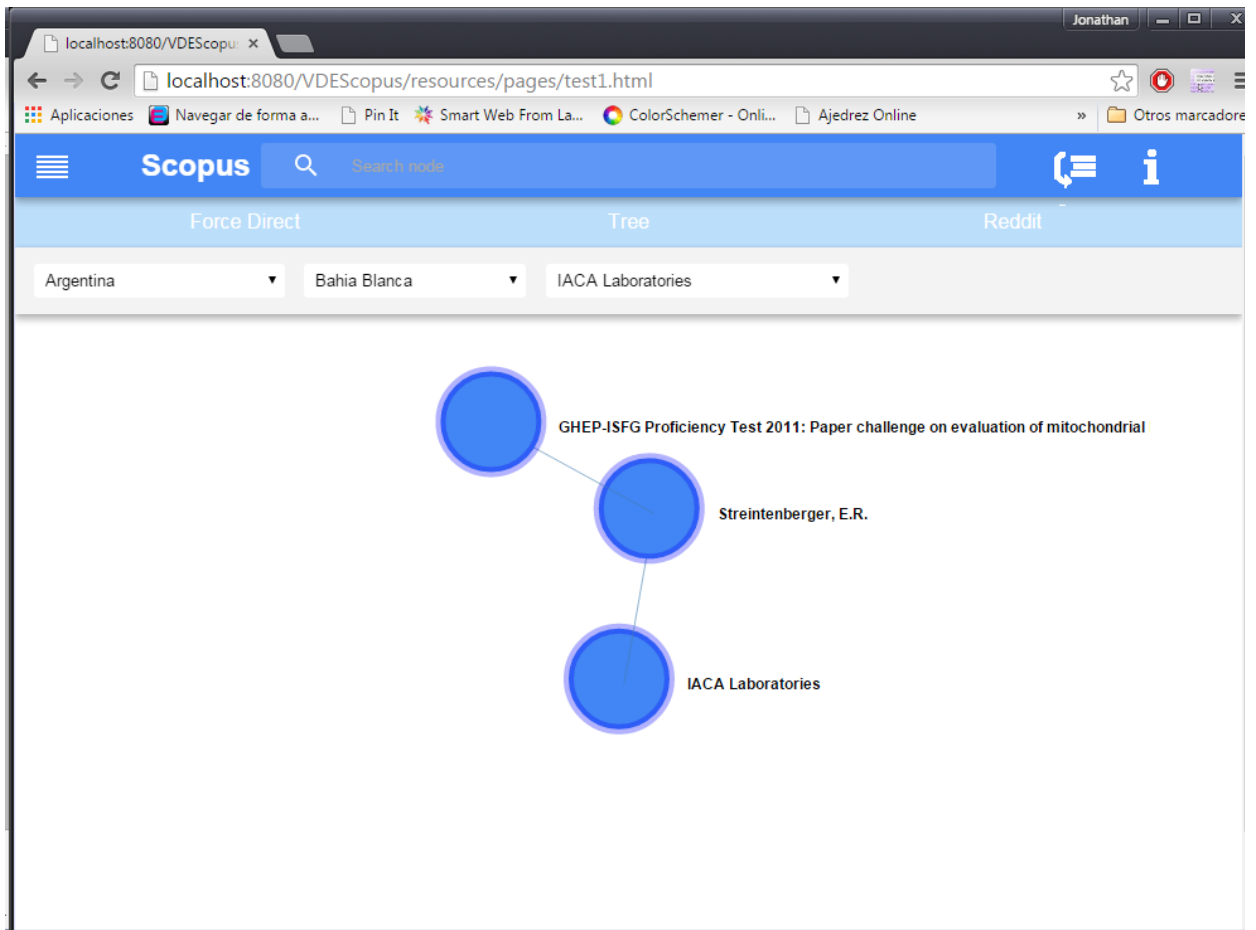


Figura 52. Tree.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015

Reddit.

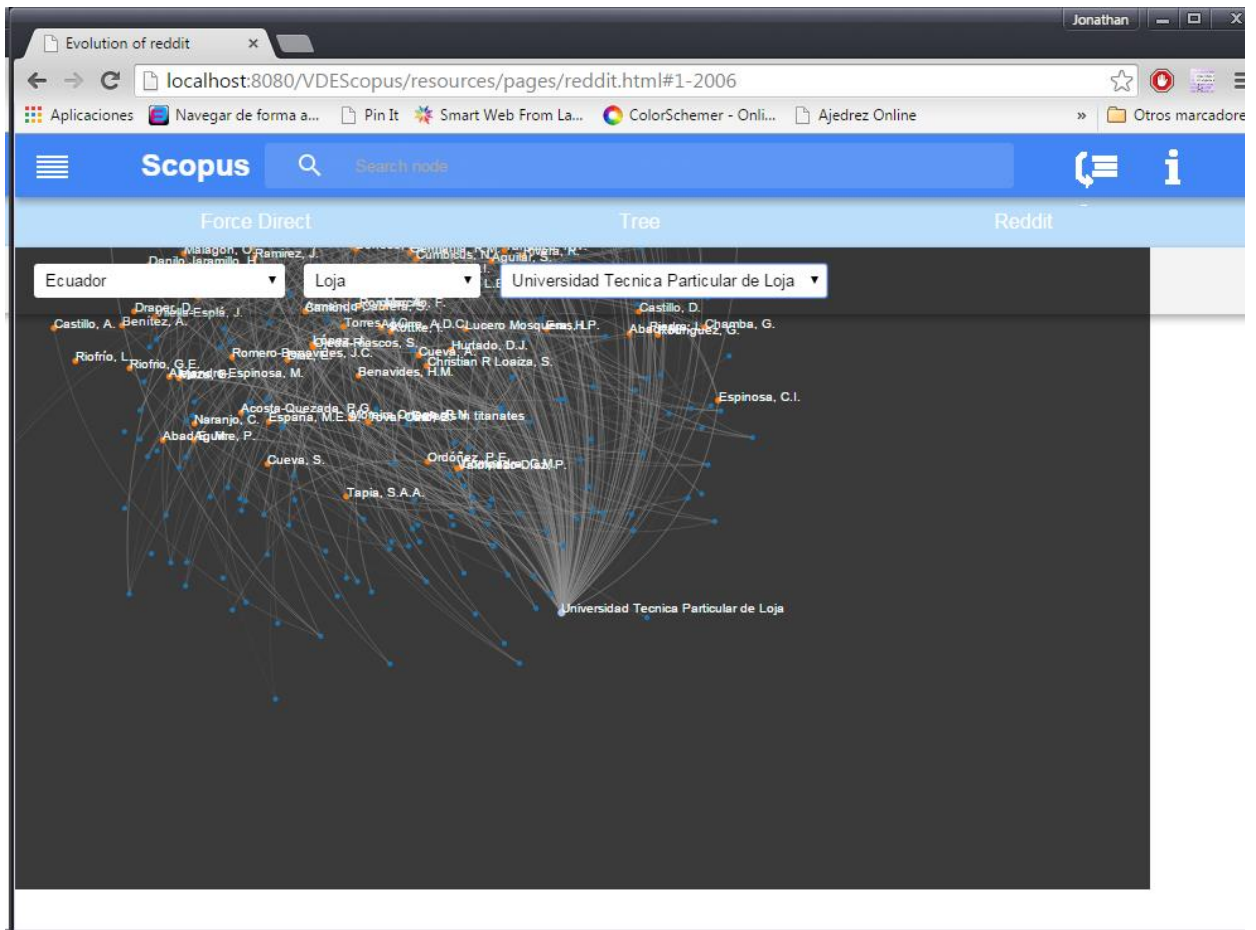


Figura 53. Reddit.
Fuente: Servicios Web VDE
Elaboración: Jonathan Yaguachi, Octubre 2015