



# UNIVERSIDAD TECNICA PARTICULAR DE LOJA

*La Universidad Católica de Loja*

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMATICOS Y COMPUTACION

Implementación de Principios de Web Semántica aplicados a Registros de la Senescyt

Trabajo de fin de titulación

**AUTORA:** Haro Vinueza Crystian Antonio  
**DIRECTOR:** López Vargas Jorge Afranio, Ing.

CENTRO UNIVERSITARIO LOJA  
2012

## Certificación

Ingeniero.

Jorge Afranio López Vargas  
DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

### C E R T I F I C A:

Que el presente trabajo, denominado: "Implementación de Principios de Web Semántica aplicados a Registros de la Senescyt " realizado por el profesional en formación: Crystian Antonio Haro Vinueza; cumple con los requisitos establecidos en las normas generales para la Graduación en la Universidad Técnica Particular de Loja, tanto en el aspecto de forma como de contenido, por lo cual me permito autorizar su presentación para los fines pertinentes.

Loja, Octubre de 2012

f) .....

## CESIÓN DE DERECHOS

Yo, **Crystian Antonio Haro Vinueza**, declaro ser autor del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 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 que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

-----  
**Crystian Antonio Haro Vinueza**

## AUTORÍA

Las ideas, opiniones, conclusiones, recomendaciones y más contenidos expuestos en el presente informe de tesis son de absoluta responsabilidad del autor.

.....  
**Crystian Antonio Haro Vinuesa**

## **DEDICATORIA**

A mi esposa Azucena que es mi inspiración

A mis hijos Catalina y Matías

A mis Padres y hermanos que siempre me han apoyado.

## **AGRADECIMIENTO**

A Dios que siempre me guía

A mi esposa que es mi vida

A mis hijos por ser la razón de lucha

A mis padres que me inculcaron valores cristianos

A mis hermanos en especial a mis hermanos Alex y Erika quienes han sido mis mentores

A los profesores de la UTPL por su ayuda desinteresada

A mis compañeros que siempre fueron una guía y una ayuda

A todas las personas y organizaciones sin fines de lucro que han puesto información que me ha servido de guía de aprendizaje en este largo camino.

# Contenido

<b>CESIÓN DE DERECHOS</b> .....	<b>III</b>
<b>AUTORÍA</b> .....	<b>IV</b>
<b>DEDICATORIA</b> .....	<b>V</b>
<b>AGRADECIMIENTO</b> .....	<b>VI</b>
<b>CAPITULO 1. Web Semántica</b> .....	<b>1</b>
1.1. <i>Introducción</i> .....	1
1.2. <i>Fundamentos</i> .....	2
1.3. <i>Historia</i> .....	2
1.4. <i>Bases</i> .....	3
1.4.1. <i>RDF</i> .....	4
1.4.2. <i>OWL</i> .....	4
1.4.3. <i>RDFS</i> .....	5
<b>CAPITULO 2. Repositorios RDF</b> .....	<b>6</b>
2.1. <i>Repositorios RDF</i> .....	6
2.1.1. <i>Diferencias entre DBMS y repositorio RDF</i> .....	6
2.2. <i>SPARQL</i> .....	7
<b>CAPITULO 3. Vocabularios Empleados</b> .....	<b>9</b>
3.1. <i>Breve descripción del problema</i> .....	9
3.1. <i>Una posible solución</i> .....	10
3.2. <i>Vocabularios Base</i> .....	11
3.1.1. <i>Vocabulario FOAF</i> .....	11
3.1.2. <i>Vocabulario DC</i> .....	12
3.1.3. <i>Vocabulario SKOS</i> .....	12
3.1.4. <i>Vocabulario SIOC</i> .....	13
3.1.5. <i>Vocabulario EUROPEANA</i> .....	13
3.1.6. <i>Vocabulario RDFS</i> .....	13
<b>CAPITULO 4. Visualización de Información</b> .....	<b>18</b>
4.1. <i>Visualización de Metadatos</i> .....	18
4.1. <i>Visualización de metadatos con SPARQL</i> .....	18
4.2. <i>Plugins o herramientas para generar gráficos</i> .....	19
4.2.1. <i>Google Chart API</i> .....	19

4.2.1.1.	Tipos de Gráficas .....	19
4.2.1.2.	Personalización.....	19
4.2.2.	<i>PREFUSE</i> .....	20
4.2.2.1.	Tipos de Gráficas .....	20
4.2.2.2.	Personalización.....	21
4.2.3.	<i>D3</i> .....	21
4.2.3.1.	Tipos de Gráficas .....	22
4.2.3.2.	Personalización.....	22
4.2.4.	<i>RAPHAEL</i> .....	22
4.2.4.1.	Tipos de Gráficas .....	23
4.2.4.2.	Personalización.....	23
4.3.1.1.	Vista “Registro de títulos” .....	25
4.3.1.2.	Vista “Totales por Provincia” .....	26
4.3.1.3.	Vista “Titulados SENESCYT” .....	27
4.3.1.4.	Vista “Títulos por Provincia” .....	28
4.3.1.5.	Vista “Títulos” .....	29
4.3.1.6.	Grafico “Total de Títulos por Provincia”.....	30
4.3.1.7.	Grafico “Total de Títulos por Provincia en barra horizontal”.....	30
4.3.1.8.	Grafico “Porcentaje de Títulos por Provincia”.....	31
4.3.1.9.	Grafico “Porcentaje de Profesionales por Título”.....	32
4.3.1.10.	Grafico “Top 25 de las carreras más seguidas.”.....	32
<b>CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>34</b>
6.1.	<i>Conclusiones Generales</i> .....	34
6.2.	<i>Recomendaciones</i> .....	35
6.3.	<i>Casos de uso</i> .....	36
<b>Anexo .....</b>		<b>37</b>
<b>Glosario .....</b>		<b>53</b>
<b>Índice Alfabético .....</b>		<b>58</b>
<b>Bibliografía .....</b>		<b>60</b>

## INDICE DE TABLAS Y FIGURAS

Tabla 1. Características funcionales de los repositorios.....	7
Figura2. Core RDFS de la implementación. ....	16
Figura3. Ejemplos de salidas desde SPARQL Endpoint. ....	17
Tabla2. Tabla resumen del Vocabulario de la Implementación. ....	17
Figura4. Imágenes representativas de Google chart API.....	20
Figura5. Imágenes representativas de PREFUSE.....	21
Figura6. Imágenes representativas de D3 .....	22
Figura7. Imágenes representativas de Raphaël.....	23
Figura8. Vista “registro de títulos”.....	25
Figura9. Vista “Título y registro del nodo”.....	26
Figura10. Vista “Titulados SENESCYT”.....	27
Figura12. Vista de Títulos por Provincia.....	28
Figura13. Vista de Títulos.....	29
Figura14. Grafico Total de Títulos por Provincia.....	30
Figura15. Grafico Total de Títulos por Provincia hbar.....	31
Figura16. Grafico Porcentaje de Títulos por Provincia.....	31
Figura17. Grafico Porcentaje de Profesionales por Título.....	32
Figura18. Grafico Porcentaje de Profesionales por Título.....	33
Figura19. Ventana principal del OUTWIT.....	38
Figura20. Arquitectura de la implementación.....	40
Figura21. Panel de PHPMYADMIN opción de crear base de datos.....	42
Figura22. Opción de tipo de instalación.....	43
Figura23. Ventana de configuración de base de datos.....	43
Figura24. Grafico del home del sitio.....	44
Figura25. Administración de módulos en DRUPAL.....	44
Figura26. Instalación de módulos a través de directorio.....	45
Figura27. Estructura de las tablas del Repositorio.....	47
Figura28. Módulo de importación del PHPMYADMIN.....	47
Figura29. Nodos agregados resultado de la reingeniería.....	48
Figura30. Opción SPARQL Endpoints Registry.....	49
Figura31. SPARQL Endpoints añadidos al sitio.....	49
Figura32. Configuración de tipos de contenidos del endpoint.....	50
Figura33. Campos añadidos al registro del endpoint.....	50
Figura34. Mapeos RDF para cada tipo de contenido.....	51
Figura35. Endpoint de SPARQL.....	51

## RESUMEN EJECUTIVO

La presente investigación está enfocada en la aplicación de principios de Web Semántica, los cuales permitan consultar información referente a los títulos académicos de los profesionales ecuatorianos registrados en la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e innovación (SENESCYT), esta servirá de modelo para nuevas investigaciones que tengan como finalidad ampliar los horizontes de las consultas web y creación de vocabularios semánticos, esta propuesta tecnológica brindará a sus usuarios beneficios tales como: mejoramiento en la recopilación de datos, rapidez y agilidad en las consultas de datos, además de lograr con esto una estructuración más exacta de los datos por medio de la aplicación de principios semánticos.

Para seleccionar las herramientas semánticas que se utilizarán en la investigación, se evaluó varias de las disponibles en la actualidad, tomando en consideración factores como: características, funcionalidad, soporte, disponibilidad y compatibilidad, todos estos aspectos relevantes al momento de comparar productos. Por lo tanto, las herramientas semánticas que mejor se adaptaron a este proyecto fueron las que se apegaron a los requerimientos y necesidades, respetando estándares de diseño, así como de lenguajes empleados.



# CAPITULO 1. Web Semántica

---

## Conceptualizandola www

### 1.1. *Introducción*

La web como la conocemos normalmente es un vertiginoso e imparable concepto que día a día gana más adeptos gracias a un constante y casi sin límite flujo de recursos e innovación. Una herramienta que se crea a finales del siglo 20 y que ha sido la industria de mayor crecimiento en el mundo y que por supuesto no es difícil darse cuenta del porqué de este crecimiento cuando cerca del 90% de la población mundial la conoce y dos tercios la han utilizado o la utilizan frecuentemente. Con un millar de herramientas, portales, páginas, redes sociales y otros recursos más que la componen. La WWW ha creado en las personas esta adicción por estar en la red, por ser parte de ella, por compartir su conocimiento y es que la red es el mundo y el mundo hoy es la www.

A pesar de todo este impresionante despliegue humano y monetario global que es parte de la web, es imposible que abarque la totalidad de aspectos que esta red debería cubrir. Por citar uno de los aspectos que saltan a la vista a cualquier usuario de internet es el hecho de que los buscadores son máquinas aún muy restringidas que en muchas ocasiones no presentan la información que se busca. Si hace un pequeño experimento y digita "buscador google" desde el navegador por excelencia Google recibirá como resultado muchas opciones en respuesta a la búsqueda, volviendo la respuesta ambigua a una pregunta concreta que debería arrojar un resultado concreto y exacto. Y es que los motores como google buscan enlaces a páginas que contengan una o varias de las palabras que ingresamos, con mecanismos como la indexación para acelerar las búsquedas y encontrar respuestas en el menor tiempo posible, o en la actualidad poniendo etiquetas a los sitios para tenerlos almacenados y poder encontrarlos y referenciarlos de manera más eficiente. Pero la eficiencia no es sinónimo de éxito pues las respuestas a nuestras preguntas resultan ser ambiguas, la web semántica ayuda en este sentido aportando esta capa de significados a la estructura de los contenidos, los hacen inequívocos y únicos para el usuario y sobre todo para las máquinas que procesan estos datos quitando así estas ambigüedades por medio de identificadores únicos.

Esta lógica que aporta la semántica es algo parecido a que google contratara a millones de personas para las billones de consultas diarias y estas personas leyeran nuestras consultas, las entendieran y nos dieran una respuesta acertada de este requerimiento eliminando así esta ambigüedad en los resultados. Esto es lo que se pretende con la web semántica, ese entendimiento semántico entre personas y máquinas.

## **1.2. Fundamentos**

La web semántica tiene ya algunos años y algunas investigaciones todavía no ha alcanzado un sitio privilegiado en la web, no se ha implementado como un estándar web por no romper la estructura actual o tal vez por miedo a dejar de ocupar los sitios que ocupan las grandes compañías de internet, u otras razones particulares que han dejado el tema semántico de lado.

Algunos investigadores de Inteligencia artificial que han participado en proyectos semánticos, concluyen que los sistemas basados en el conocimiento constan de dos partes fundamentales, que son la base del conocimiento y el motor de inferencia lógica, pero que las aplicaciones de web semántica pueden tener diferencias en estos fundamentos.

Las aplicaciones de web semántica construyen sus bases de conocimiento uniendo datos de otras aplicaciones semánticas, que compartan su conocimiento con las primeras, pudiendo ser este conocimiento compartido un vocabulario semántico, una herramienta como SPARQL o un repositorio RDF. La Web semántica permite crear vocabularios con estandarización que a la vez no sean una plantilla para uso universal sino más bien un modelo de datos, el llamado modelo RDF.

Este modelo está compuesto por tres partes que son: el sujeto que es el tipo de recurso referido, el predicado que identifica una cualidad del recurso: blanco, negro, dueño, etc. y el objeto que es el valor del recurso por ejemplo "Crystian Haro". Estas partes del modelo se identifican mediante identificadores universales URIS. Todo este contenido relevante para la unicidad de los recursos no es suficiente para formar la web semántica y las máquinas que busquen metadatos en ella, es necesario que se tenga un vocabulario y propiedades particulares que se utilicen de ellos y que formen un todo en conjunto, como es resumido de [1].

## **1.3. Historia**

El concepto de semántica como tal nace en la década de los setenta cuando Allan M. Collins, científico americano que al trabajar con proyectos de inteligencia artificial para escolares plantea la teoría de la memoria semántica y cómo esta interactuaría con un sistema experto.

Tras diversas publicaciones de Collins y otros científicos connotados nace en sí una idea nueva que sirve para representar el conocimiento semántico estructurado, que se entiende como una forma de aportar lógica a los datos y que luego se extiende este concepto hacia un ámbito en el que se puedan consumir estos datos y utilizar agentes para hacerlo. Y que estos agentes puedan analizarlos datos sin necesidad de ayuda humana.

La idea de la semántica aplicada a la red es decir la "Web Semántica", la plasma su creador y representante máximo Tim Berners Lee que además es inventor de la WWW y el W3C. Berners Lee crea y define a esta nueva red semántica como "la red de los datos que pueden ser

procesados directamente o indirectamente por máquinas”, y su propósito radica en facilitar al usuario el poder combinar información de manera más eficiente y rápida. Para poder lograr este objetivo, necesitamos de una máquina que pueda interpretar de manera similar nuestras necesidades y plasme nuestras búsquedas como tuyas y comprender de manera parecida al humano lo que se pretende encontrar.

Gracias a la ayuda de HTMLha sido posible en parte presentar estos metadatos como etiquetado o adjuntarlas como parte de un XML, y aunque el potencial del HTML es inmenso todavía se limita a recorrer las etiquetas y no las interpreta. Para la solución de estos problemas se piensa en una versión semántica de este lenguaje que no solo especifique los detalles del diseño de la página sino que interprete los conceptos de las etiquetas,convirtiéndose en una máquina de interpretación semántica, como refiere Wikipedia en[2].

#### **1.4. Bases**

La web semántica es una arquitectura que intenta que la web tradicional se convierta en una base de datos global, con la que las máquinas puedan trabajar directa y automáticamente. Para este propósito la web semántica debe poseer principios técnicos que tienen que regir su funcionamiento. Estos principios son sencillos y claros, en primer lugar se necesita que la información contenida en una web semántica posea referencias de lectura mecánica, segundo debe tener interpretación global sin importar el idioma en que este hecha, además de poseer documentos que muestren las relaciones de los datos que contiene y por ultimo debe estar basada en una estructura de datos que permita ser creada e interpretada por las máquinas, estructura basada en recomendaciones relacionadas con la web semántica, recomendaciones que usan tecnologías como XML, RDF y OWL, como sustenta Garland Foster en[3].

XML como vínculo de intercambio de información,independiente de la aplicación que añade la sintaxis para documentos estructurados, formando jerarquías con nombres y atributos, que además no posee restricción semántica de nombres o atributos pudiendo ser leído por personas o máquinas.

RDF “como el marco referencial para la descripción de metadatos convirtiéndose en un lenguaje para representar la web” a través de esquemas, como sustentan Brickley y Guha en [28]. Esquemas que pueden ser clasificados para un dominio específico, mediante el uso de namespaces que apunten a esquemas externos como en el caso de este proyecto.RDF permite interpretar los recursos para que puedan ser accesibles desde cualquier lugar sin importar el vocabulario que utiliza. RDF, además utiliza las tripletas para asignación de valores de los metadatos.

OWL como ampliación del vocabulario de RDF, añadiendo más vocabulario para describir características de clases y relaciones, proporcionando descripciones de relación entre la información de otras web semánticas, como sustenta Culley Ai en [26].

### 1.4.1. RDF

RDF es en esencia un XML mejorado, pues agrega las particularidades semánticas a las etiquetas del XML haciendo mucho más rápidas las búsquedas y disminuyendo recursos para las mismas, además que vuelve el árbol XML mucho más fácil de recorrer. RDF en su composición posee los siguientes objetos:

Los recursos engloban absolutamente todo lo que podría describirse con el RDF y que se denomina un “recurso”. Pero el termino recurso a primera vista parece un concepto muy abstracto y de hecho en parte lo es ya que los llamados recursos son o pueden ser muchos objetos en la web y también fuera de ella, una página, una etiqueta Tag o una etiqueta XML que se encuentra dentro de una página, una colección de páginas o quien sabe a corto plazo hasta bloques de categorías de páginas enlazadas son recursos para RDF. También estos recursos pueden no ser web o no estar accesibles directamente como los libros y revistas inclusive documentos científicos como tesis y demás que estén contenidos en URIS para cualquier objeto que uno pueda imaginarse, como refiere W3C en[4].

#### Veamos un ejemplo<sup>1</sup>:

Consideremos la siguiente oración:

*“Luis Velasco es el creador del recurso <http://www.w3.org/home/lvelasco>”*

En este ejemplo la oración contiene las tres partes fundamentales de la sentencia:

Sujeto (recurso)	<a href="http://www.w3.org/home/lvelasco">http://www.w3.org/home/lvelasco</a>
Predicado (propiedad)	Creador
Objeto (literal)	Luis Velasco

### 1.4.2. OWL

El lenguaje de Vocabularios Web OWL, que por sus siglas en español quiere decir “lenguaje de ontologías web”, cercanamente relacionado con RDF, pues OWL brinda un aporte a RDF pues le da significado, generando inferencia lógica en el razonamiento y actúa con una sintaxis en RDFXML y es representado con tripletas RDF, además todos los documentos OWL tendrán

---

<sup>1</sup> Recopilación información de: [www.w3.org](http://www.w3.org)

todas las características de este vocabulario y el usuario podrá utilizarlas o utilizar solamente la versión lite.

Al no ser una extensión OWL permite inferencias que no se podría hacer con una tripleta simple de RDF. OWL posee sublenguajes, uno más sencillo y funcional como es OWL lite pero no menos potente que muestra su capacidad de implementación, a la vez que provee el potencial necesario para inferencias de agentes simples y de perfiles de personas y OWL Descripción Lógica o DL que proporciona la potencia necesaria para áreas del negocio o razonamiento más especializado, como el de la meteorología y otros. Estos dos sublenguajes aceptan las mismas construcciones de lenguaje, diferenciándose en el uso de sus características, así como las propiedades RDF que podría tener. OWL es parte de RDF schema trabajando con sus clases, individuos o propiedades, como refiere Lamarca María en [5].

### **1.4.3. RDFS**

Existe un lenguaje de descripción de vocabularios RDF que lo define y lo estandariza, haciendo uso correcto de términos y palabras, así como revisando que estas transmitan un exacto significado y sentido, llegando con esto a aplicaciones que procesen enunciados y puedan transmitir nuestros datos. A través de estas descripciones se condensa un esquema que viene a ser un diccionario de términos definidos que se utilizan en declaraciones y sentencias en RDF y que permiten que estas tomen un sentido correcto.

Estos esquemas otorgan significado a los términos y utilizan para ello a los Namespace que separan explícitamente los metadatos y modelan estos datos en las tripletas por medio de las clases, estos esquemas deberán seguir normas de creación, diferenciándolos universalmente de cualquier otro objeto o tripleta del mismo tipo que se encuentre en la web. RDF esquema no solo cuenta con RDFS, sino que ha creado otros vocabularios para poder diversificar su uso.

RDFS trabaja con Namespace que son especificadores de definiciones, con relación a la programación, estos Namespace trabajan como los métodos y también pueden tener sobrecarga del mismo con valores diferentes, además ofrece una estructura potente para crear nuevos vocabularios que se adapten a las necesidades específicas de cada aplicación y que se los etiqueta como RDF esquemas. Estos espacios de nombre identifican una declaración en un esquema y sirve para relacionar el uso de las palabras en el contexto del esquema, en el que una definición se encuentra. Este esquema RDF lo forma las declaraciones de propiedades y clases que utilizarán los términos definidos en las declaraciones, como sustenta Lamarca María en [6].

# CAPITULO 2. Repositorios RDF

---

## 2.1. *RepositoriosRDF*

Un repositorio en RDF es una base de datos que puede ser de variado tipo, pero que contiene las características RDF que lo ligan a él. Por ejemplo para este proyecto la base de datos es MYSQL y a través del módulo RDX se añade la característica RDF a la base, volviéndola un repositorio. Esto permite un acceso claro y sencillo a los datos almacenados a través de una tripletaRDF o grafos. Estos repositorios son los también llamados almacenes de tripletas o triple stores, como refiere Torre, González, Illarramendi y Bermúdez en [31] .

Los repositorios nos brindan el acceso a los datos almacenados como grafosRDFpara poder recuperarlos por medio de consultas de datos o en forma de archivos. Algunos almacenes de tripletas pueden usar el almacenamiento y la funcionalidad delDBMS, otros en cambio pueden implementar su propio motor de base de datos, sin usar el almacenamiento y recuperación de otro DMBSy un tercer grupo que soporta ambas arquitecturas. Algunos motores de base de datos en la actualidad están brindando ya un soporte para RDF como es el caso de ORACLE que brinda funcionalidades de declaraciones RDF, búsqueda y recuperación de datos, acceso a datos mediante SQL y SPARQL además de herramientas de visualización de los esquemas RDF creados. Los repositorios RDF están teniendo un auge muy grande y se están extendiendo en casi todos los ámbitos posibles, pues permiten manejar datos más complejos con menor costo, como refiere W3C en [7].

### 2.1.1. *Diferencias entre DBMS y repositorioRDF*

Entre las mayores diferencias a destacar entre un repositorio semántico y una base de datos tradicional están:

- El razonamiento automático de los repositorios gracias al uso de vocabularios que les sirven como esquemas semánticos.
- Los repositorios trabajan con un esquema más sencillo de modelo de datos, lo que les permite extenderse y actualizarse más rápidamente en su esquema.

Los repositoriosRDF almacenan importantes características RDF como las propiedades que poseen para que puedan ser consumidos a través de consultas de lenguajes como SPARQL, pudiendo ser creados de forma nativa es decir implementando un completo motor de base de datos que es optimizado por procesos y operaciones RDF que son independientes del DBMS, además de almacenar los datos en el sistemas de archivos, conocidos repositorios nativos son: ALLEGROGRAPH, JENA, OWLIMy algunos otros más, como refiere W3C en [7]. Otro tipo de repositoriosson losDBMS-BACKED que utilizan el almacén de datos propio del DBMS pero añaden procesos y operaciones RDF al mismo. Algunos ejemplos de este tipo son: ARC (del

cual se utiliza librerías adaptadas a DRUPAL para poder trabajar con el móduloRDFX), ORACLE11g entre otros. Podemos citar un tercer tipo de repositorio que es el híbrido que puede soportar tanto las arquitecturas tipo nativas como DBMS-BACKED, un ejemplo muy conocido es el de SESAME y VIRTUOSO. La siguiente tabla muestra características funcionales de estos repositorios, los tipos de consultas que soportan y otros aspectos fundamentales, como sustentan BernhardHaslhofer, ElahehMomeni, BernhardSchandl y Stefan Zander en [8].

Características funcionales							
	Transacciones	Lenguajes De consulta	Consultas Full-Text	Versionamiento	Procedencia	Inferencia	Backup restore
<b>ALLEGROGRAPH</b>	Log de transacciones	SPARQL	Índices free text	A nivel de la base	grafo	RDFS	Si
<b>JENA</b>	No posee	SPARQL con extensiones	(Lucene)	No	grafo	RDFS, OWL	No
<b>OWLIM</b>	Si	SPARQL, SERQL	Si	No	grafo	Reglas	No
<b>ARC</b>	No posee	SPARQL, SPARQL SCRIPT	No	No	grafo	vía pluggins	Dump de la base
<b>ORACLE11g</b>	Completo soporte de transacciones	SPARQL, SQL	Si	Si	Tablas especificas	RDFS, OWL, OWL2	Si
<b>SESAME</b>	Si	SPARQL, SERQL	(Lucene)	No	grafo	RDFS	Db/File System
<b>VIRTUOSO</b>	Si	SPARQL, SQL	Si	No	grafo	RDFS, OWL	Si

**Tabla 1. Características funcionales de los repositorios<sup>2</sup>.**

## 2.2. SPARQL

SPARQL es un lenguaje de recuperación y consulta en almacenes de datos, en lo que a semántica y RDF se refiere, específicamente estructurado para fuentes de datos y data sets de carácter ontológico que se encuentren en la web. SPARQL es un lenguaje de consultas RDF que es básicamente un estándar en recuperación de metadatos, sin importar en qué tipo de base de datos se encuentra el repositorio, pues SPARQL se enfoca en las características y vocabulario semántico no en la base de datos como tal, como refiere W3C en [9].

Una ventaja importante tanto de los almacenes de datos semánticos como de este lenguaje es que las consultas pueden ser hacia múltiples almacenes de datos, lo que no ocurre con los lenguajes de datos tradicionales como SQL, convirtiéndose en un ahorro en el costo y un incremento de resultados en menor tiempo. SPARQL puede obtener resultados en consultas

<sup>2</sup>Recopilación información de: [www.europeanaconnect.eu/documents/europeana](http://www.europeanaconnect.eu/documents/europeana)

con varios formatos desalida como son: RDF/XML, JSON, texto plano y otros aumentando la utilidad de las consultas y el uso que se les puede dar a las mismas. En cuanto a sus funciones es muy robusto y posee además algunas bibliotecas externas que se pueden añadir como es el caso de SPARQLMOTION. Funciones para visualizaciones y análisis estadístico como COUNT o AVG y otras de agrupamiento como Group by se pueden usar casi con la misma estructura que en SQL, otras como Pívor que pasa las filas a columnas no se encuentran en SPARQL, una opción para poder hacer este cambio es realizar las consultas SPARQL desde el endpoint sacarla con formato XML y usar un script que suba este resultado a una tabla temporal o fija para poder así elaborar consultas de tipo SQL sobre nuestro repositorio para las funciones que SPARQL no tenga soporte. Repositorios como el de ORACLE o VIRTUOSO trabajan tanto con SPARQL y SQL para las consultas y no tienen este tipo de problemas, como refiere W3C en [9].

Entre las diferencias y desventajas de SPARQL con respecto a lenguajes como SQL esta por ejemplo en que SPARQL no posee forma de realizar tuning de las consultas a diferencia de SQL con esto reduce el espectro de SPARQL para escribir programas en DBMS. Las consultas de SPARQL se vuelven muchas veces confusas y de difícil evaluación. Otra de las Diferencias entre SPARQL y SQL es que el primero se basa en grafos dirigidos aunque como hemos visto en su Core pueden ser bases de datos relacionales, mientras que SQL se basa en consultas sobre tablas, como refiere W3Cen [10].

# CAPITULO 3. Vocabularios

---

## Empleados

---

### Mapeos RDF

#### 3.1. *Breve descripción del problema*

Al inicio, la obtención de documentos en la Web constituía una enorme cantidad de texto, imágenes, multimedia, sin sentido para un ordenador en donde las personas extraían e interpretaban la información relevante, haciendo de esta red una herramienta de poca ayuda para la investigación. En la última década y debido al enorme crecimiento de la información en Internet es imposible que una persona realice estos procesos de manera rápida y eficiente, de modo que han surgido nuevas tecnologías que mejoran la gestión y recuperación de la información, aspectos como la presentación de documentos se realiza independientemente del contenido, dando paso a la era semi-automática de documentos Web. En la actualidad la WEB se ha convertido en un gigantesco medio para acceder al conocimiento, entretenimiento y los negocios, entre otras actividades económicas con y sin fin de lucro. Para esto se han desarrollado todo tipo de tecnologías que logran llevar estas actividades hasta el internauta, logrando un más fácil y sencillo nivel de uso, en el que se puede vincular bases de datos, servicios, e incluso voz y video. Aun cuando toda esta información está al alcance, para los motores de búsqueda es muy difícil todavía para estos el saber exactamente el resultado que se espera obtener, como sustentan Marlon Melo e Ilma Bonilla en [29].

Las páginas que se utilizan hoy en día están basadas en hipertexto, que permite dirigirse hacia lugares de la propia página o hacia referencias externas, facilitando la comprensión del texto y del contenido de la misma. Pero la facilidad de navegar dentro o fuera de la página para poder encontrar fragmentos de información, se vuelve muy lenta y de poca utilidad si se debe recurrir a un gran número de documentación, que requiere del lector un arduo trabajo de investigación y selección, como sustenta ecarhoh en [30].

Lograr que los equipos comprendan el significado, la semántica, de los textos y de las páginas Web es uno de los aspectos fundamentales del desarrollo que debería tener la web actual y es la principal característica de la Web Semántica. La principal diferencia de la Web semántica con respecto a la Web que conocemos hasta hoy radica en que va más allá de la publicación de documentos, implica el que cualquier dato sea susceptible de ser expuesto y reutilizado, por ejemplo contenido multimedia, metadatos, entre otros, como sustenta Blanco Sergio en [27].

La WEB semántica facilita la obtención de la información deseada de forma más eficaz, pues se basa en la idea de añadir metadatos semánticos y ontológicos al contenido. Los metadatos pueden ser definidos como “datos sobre datos”, esto quiere decir “un dato acerca de un dato”,

los metadatos son información relevante para las máquinas de gestión de la información, que permita al gestor leer entender y calibrar la información contenida en un texto, para que este logrediscernir entre el texto que debe ser mostrado en función del contexto que recibe, como sustenta ecarhoh en [30].

La WEB semántica se propone romper los límites de los esquemas actuales mediante la descripción explícita del significado, dotando a los recursos de semántica explícita comprensible para las máquinas, dotando a cada palabra de información estructurada que será registrada en campos, convirtiendo el texto plano en bases de datos relacionales, como sustenta ecarhoh en [30].

### **3.1. *Una posible solución***

Para lograr que los máquinas entiendan el significado del contenido de las páginas necesario añadirle lógica al mismo haciendo que estos datos posean atributos propios volviéndose objetos únicos en la red. Partiendo de esta premisa se construye una aplicación con componentes semánticos que añaden lógica a la página, tomando datos extraídos de una fuente externa como es el caso de la SENESCYT, datos que son registros de títulos que la entidad gubernamental tramita y que pueden ser revisados libremente en su página.

Se construye un repositorio RDF para los datos extraídos, además de un vocabulario semántico, se añade SPARQL como lenguaje de consulta semántica para los datos que se transformaran a RDF, se configura un SPARQL Endpoint para poder poner nuestros metadatos disponibles a cualquier posible consumidor y se realizan visualizaciones para mostrar los metadatos obtenidos.

Para el proceso de adquisición de datos, se analiza algunas herramientas de extracción y minería de datos enfocadas a la semántica y que permitirán obtener de manera rápida la información requerida para poderla tomar como muestra en el proyecto, se escoge una herramienta llamada OUTWIT y mediante este software de extracción de datos conseguir una muestra representativa de 44.000 registros, la cual es segmentada por apellidos.

Los registros extraídos se depuran en su formato, características y calidad de la data, para que pudieran estar conforme a la estructura del vocabulario que se creó. La depuración de los registros es un proceso pre y posterior a la toma de la data, en la etapa preliminar a la extracción de datos se depura el formato y características y en la etapa posterior a la extracción de datos, la depuración de los registros se enfoca en buscar registros repetidos, vacíos, o incomprensibles.

Posterior a la depuración de los datos, estos son insertados en las tablas referentes en la base de datos relacional utilizada y conforme a los campos creados en el vocabulario semántico del proyecto, que a su vez es parte del repositorio y por este proceso de inserción son transformados en tripletas RDF, transformando así los datos simples en metadatos.

### **3.2. VocabulariosBase**

Para esta implementación se han utilizado algunos vocabularios semánticos tales como:FOAF, DC, SKOS, SIOC, EDM y RDFS,utilizando sus propiedades para crear un vocabulariopara esta aplicación ycorrespondiéndolas con los datos que se pretende transformar a RDF. Las propiedades utilizadas de los vocabularios antes dichos, concuerdan perfectamente con los campos de los registros de la Senescyt que se pretende enriquecer, así por ejemplo propiedades como las del vocabulario FOAF: name o title, son campos perfectos para referenciar los datos de: Nombre y Título. Otros vocabularios como DC y SIOC pueden tener propiedades parecidas entre sí como para mencionar un nombre o una ubicación y pueden ser utilizados para referenciar un atributo.

El uso de un vocabulario común ya establecido para agregar conocimiento a los datos no está separado de los principios semánticos que se pretende aplicar en el proyecto, más bien nos sirve de ayuda para una correcta utilización en cuanto a vocabularios y su uso es parte de la aplicación de principios semánticos a los registros extraídos, estos registros del SENESCYT son datos de personas que sirven como muestra para aplicar en ellos las propiedades de los vocabularios antes mencionados y lograr que estos se enriquezcan semánticamente y se conviertan en metadatos, que puedan ser utilizados por cualquier otro sitio, usuario o máquina que los quiera consumir, o podrían crear también vocabularios propios creando sus correspondientes propiedades y clases, siguiendo siempre los estándares de vocabularios de la W3C.

#### **3.2.1. Vocabulario FOAF**

FOAF describe perfectamente a una persona y sus intereses y una diversidad de aspectos que este vocabulario puede cubrir en cuanto a información personal, como a quien conoce una persona, de quien es amigo, que nombre de cuenta tiene o cuál es su página y sus proyectos, todo un abanico de información personal captada en un solo archivo FOAF.

FOAF es uno de los vocabularios más utilizados en lo que a aplicaciones de semántica se refiere, pues su difusión, amplio espectro de utilización y fácil aplicación, han hecho de este vocabulario uno de los más referenciados y utilizados en aplicaciones semánticas. De uso libre FOAF es el diccionario con mayor gama de propiedades y clases que existe, un proyecto creado por la W3C encaminado 100% a la gente, como si de un pequeño perfil se tratara, en el cual se pueden mostrar mucha información acerca de una persona, convirtiéndose en una muestra de lo que serán las herramientas colaborativas en un futuro cercano. Redes sociales, de descripción, de representación o toda red que pretenda dar una descripción simplificada de un recurso podría hacer uso de este vocabulario y así mejorar el manejo de la información que contiene, así como una mejor forma de consumir los datos presentes en el repositorio.

Simple y sencilla podría decirse que es la sintaxis de FOAF ya que en esencia es un documento de texto que podría estar con un formato RDF, JSON, que tiene como final macenar

las clases que puedan representar por ejemplo a una persona, representar la propiedad de un nombre o representar un documento de identidad. Otra característica a resaltar de FOAF es su cualidad de enlazar un archivo FOAF con otro con facilidad, característica con visión a futuro que ayudará a los autómatas a la recolección de datos, como sustenta W3Cen [11].

La vinculación de los archivos FOAF resulta sencilla insertando marcas en el head de la página HTML, manejando el estándar de nombre de archivo FOAF.RDF. En el caso de DRUPAL el vocabulario FOAF se importa como un archivo .RDF y DRUPAL con su módulo RDFX lo reconoce como tal no necesitando insertar etiquetas en el head de la página para sus referencias. Estas características identificativas de las propiedades del vocabulario FOAF tanto como las cualidades que permiten que cada individuo o agente del cual se habla sea único han hecho justificable su utilización para campos como nombre, título y provincia.

### **3.2.2. Vocabulario DC**

Este vocabulario es uno de los más utilizados en la actualidad en muchos sectores de interés, que puede ser utilizado con HTML, XML y RDF. El Dublín Core posee pocas propiedades para usar, alrededor de quince a diferencia del vocabulario completo DCMI, pero al ser sus elementos tan genéricos y simples pueden utilizarse para referir una gama muy amplia de recursos, además que sus propiedades se pueden repetir y pueden aparecer en cualquier orden ayudando con esto a la elaboración de un vocabulario más fácil de construir. Dublín Core se convirtió en estándar en 2003 y se lo utiliza en diferentes estamentos gubernamentales y privados del mundo, Organizaciones educativas y sectores científicos de la investigación, como refiere Wikipedia en [12].

Estas características estandarizadas y potentes justifican su utilización.

### **3.2.3. Vocabulario SKOS**

El vocabulario SKOS nos proporciona una ayuda para representar esquemas conceptuales, taxonomías, tesauros, esquemas de clasificación además de otros vocabularios, como sustenta Pastor José en [13].

SKOS es una manera de migrar a semántica antiguos sistemas del conocimiento con bajo costo, adicionalmente permite crear nuevos sistemas de organización debido a que posee un sencillo lenguaje de modelado. Implementado solo o junto con otros lenguajes más formales como OWL aportando la parte informal de lenguajes formales como OWL SKOS demuestra ser una herramienta de ayuda para aplicaciones semánticas del tipo web social.

La principal función de este vocabulario es la de ayudar a que varios vocabularios conceptuales compartan el mismo espacio de acción, permitiendo ser reutilizado. Esta capacidad para compartir información y combinarse con otros vocabularios justifica el uso de SKOS en la presente implementación, como refiere Pastor José en [14].

### **3.2.4. Vocabulario SIOC**

SIOC es un vocabulario que proporciona métodos para interconectarse a redes sociales y blogs, este vocabulario se define utilizando un esquema RDFS y puede usar otras tecnologías como FOAF para enriquecer el vocabulario.

EL vocabulario SIOC está basado en RDF y trabaja tanto con RDFS y OWL, pudiendo ser capaz de leer información contenida en internet en diversas plataformas como blogs y sistemas de gestión de contenidos, comunidades online wikis y otras que se enfocan en un tema específico, básicamente se encamina a la construcción de una red que englobe todas estas redes sociales y usualmente se lo usa en conjunto con FOAF y SKOS como en nuestro proyecto, como refiere W3Cen [15]. Y esta interacción de SIOC con redes sociales hace muy justificable su inserción en la presente implementación.

### **3.2.5. Vocabulario EUROPEANA**

Europeana es un vocabulario centrado en representar metadatos de objetos culturales, suministrando la posibilidad de representarlos digitalmente. Europeana se ubica en un contexto de agregación de datos, donde los objetos pueden ser complejos y en el que diferentes proveedores de datos pueden alojar diferentes visiones de los mismos. EDM reutiliza, amplía y ha sido incorporado por otros conjuntos de elementos como sustentan Antoine Isaac, et al. en [25].

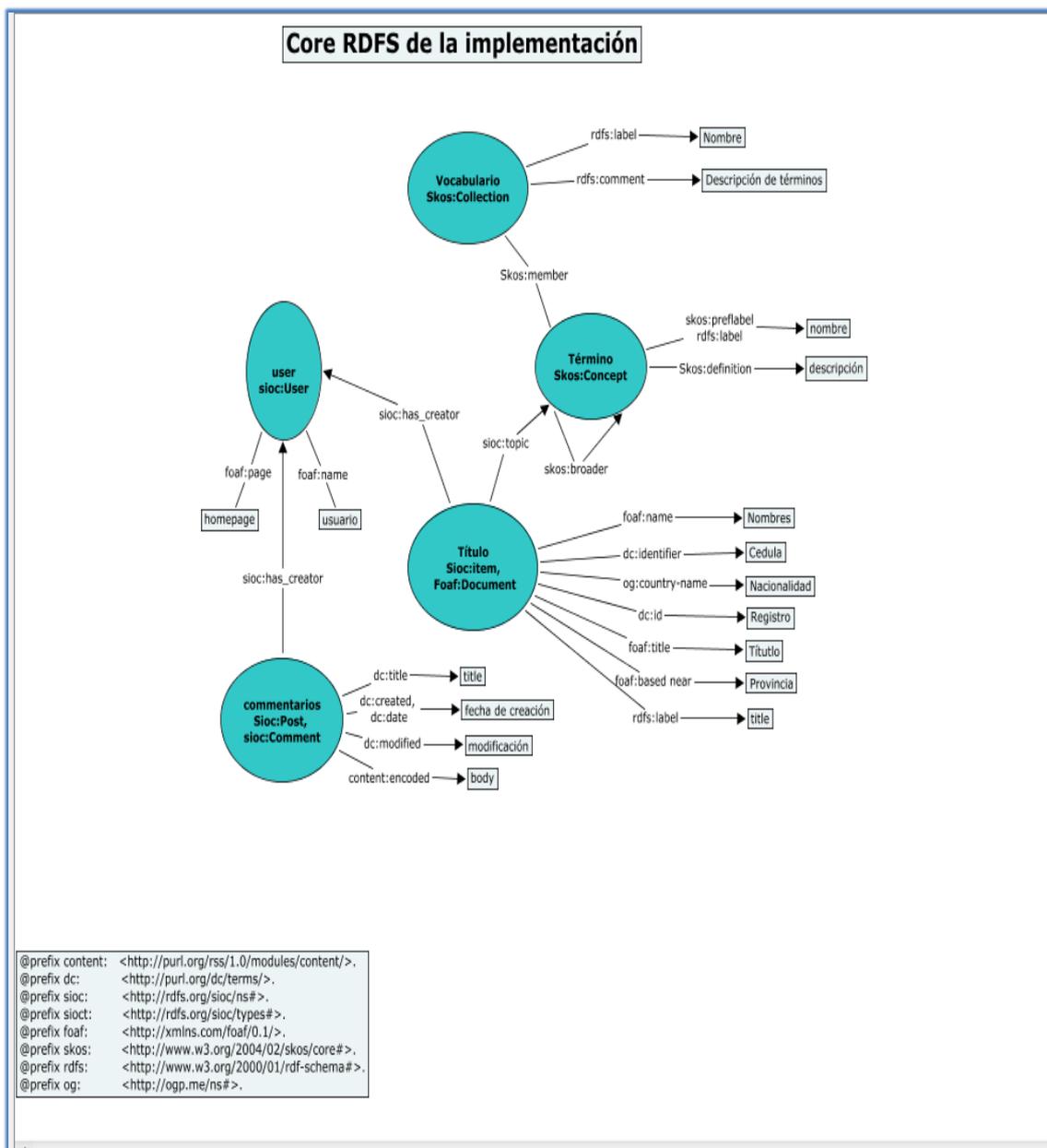
### **3.2.6. Vocabulario RDFS**

El vocabulario RDFS esquema proporciona al RDF tradicional mecanismos de especificación de clases y propiedades y como estas se relacionarán. Permite la jerarquización de las clases y que los recursos se definan como instancias de una clase. La ayuda para encapsular las tripletas RDF en paquetes llamados recursos hace que se justifique el uso de este vocabulario.

Existen muchos otros vocabularios en el mercado incluso podemos crear nuestros propios vocabularios por medio de un editor de texto o por medio de herramientas especializadas para el efecto como Odontowiki, knoodl y otros más, pero estos requieren un gran conocimiento de RDF y además suponen gran trabajo en lo concerniente a programación y estructuración de los mismos y este proyecto pretende utilizar herramientas al alcance que permitan llevar a cabo el objetivo del mismo y utilizando vocabularios existentes se lo logra, por ejemplo utilizando la propiedad FOAF:TITL se logra un concepto claro y entendible del campo título sin necesidad de crearlo mediante un vocabulario propio, como refiere García Grecia en [16].

### **3.2.7. Obtención del RDFS del vocabulario desarrollado**

Los vocabularios anteriormente referidos han servido de base para la implementación del presente proyecto, siendo utilizadas sus propiedades y clases para la creación de un tipo de recurso llamado "título", que es el recurso utilizado para dar vida semántica a los registros de la SENESCYT. El siguiente grafico se ha desarrollado gracias a la herramienta cmaptools creada y referida por HCM (2010), que muestra el esquema RDF de la implementación realizada mostrando las clases, propiedades y recursos de las mismas así como los vocabularios empleados y sus Namespace.



**Figura1. Vista completa del RDFS de la implementación<sup>3</sup>.**

<sup>3</sup> Diagrama generado con la herramienta Cmaptools, disponible en: [cmap.ihmc.us/](http://cmap.ihmc.us/)

Esta figura muestra una vista completa del CORE RDFS de la implementación, es decir el esquema implementado no solo en el vocabulario sino para toda la aplicación.

Del esquema anteriormente mencionado también se puede obtener el archivo del vocabulario en formato TURTLE, archivo que se muestra en la siguiente gráfica.

**Archivo Turtle del esquema**

```

@prefix Skos: <http://localhost/Skos#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix sioc: <http://localhost/sioc#>.
@prefix content: <http://localhost/content#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://localhost/foaf#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix skos: <http://localhost/skos#>.
@prefix dc: <http://localhost/dc#>.
@prefix daml: <http://www.daml.org/2001/03/daml+oil#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://localhost/default#>.
<http://localhost/comentarios%20Sioc#Post,%20%20sioc:Comment>
  content:encoded :body ;
  <http://localhost/dc#created,%20dc:date>
  <http://localhost/default#fecha%20de%20creaciÃ³n>
    dc:modified :modificaciÃ³n ;
    dc:title :title ;
  sioc:has_creator <http://localhost/user%20sioc#User> .
<http://localhost/TÃ­tulo%20Sioc#item,%20%20Foaf:Document>
  rdfs:label :title ;
  sioc:id :Registro ;
  dc:identifier :CÃ©dula ;
  foaf:based_near :Provincia ;
  edm:country :Nacionalidad ;
  foaf:name :Nombres ;
  foaf:title :TÃ­tulo ;
  sioc:has_creator <http://localhost/user%20sioc#User> ;
  sioc:topic <http://localhost/TÃ©rmino%20Skos#Concept> .
  <http://localhost/user%20sioc#User>
    foaf:name :usuario ;
    foaf:page :homepage .
  <http://localhost/Vocabulario%20Skos#Collection>
  rdfs:comment <http://localhost/default#DescripciÃ³n%20de%20tÃ©rminos> ;
  rdfs:label :Nombre ;
  Skos:member <http://localhost/TÃ©rmino%20Skos#Concept>.
  <http://localhost/TÃ©rmino%20Skos#Concept>
    Skos:definition :descripciÃ³n ;
  skos:broader <http://localhost/TÃ©rmino%20Skos#Concept> ;
  <http://localhost/skos#preflabel%20rdfs:label>:Nombre.

```

Figura2. Archivo TURTLE del esquema completo de la implementación<sup>4</sup>.

Se puede obtener desde el SPARQLENDPOINT de la implementación, salidas en formatos TURTLE, RDF/XML, JSON, XML y otras más, utilizando consultas SPARQL. La siguiente gráfica muestra un par de ejemplos de este tipo.

```
CONSULTA SPARQL
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
SELECT ?name
WHERE
{ ?x foaf:name ?name
FILTER regex (?name, "crystian antonio haro vinueza")
}
LIMIT 10

SALIDA EN JSON
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <!-- query time: 0.0674 sec -->
    <variable name="name"/>
  </head>
  <results>
    <result>
      <binding name="name">
        <literal>crystian antonio haro vinueza</literal>
      </binding>
    </result>
  </results>
</sparql>

SALIDA EN RDF/XML
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <!-- query time: 0.0184 sec -->
    <variable name="name"/>
  </head>
  <results>
    <result>
      <binding name="name">
        <literal>crystian antonio haro vinueza</literal>
      </binding>
    </result>
  </results>
</sparql>

SALIDA EN SERIALIZED PHP
a:2:{s:6:"result";a:2:{s:9:"variables";a:1:{i:0;s:4:"name";s:4:"rows";a:1:
{i:0;a:2:{s:4:"name";s:29:"crystian antonio haro vinueza";s:9:"name type";s:7:"literal";}}}
s:10:"query_time";d:0.0118520259857177734375;}
```

<sup>4</sup> Archivo generado con la herramienta Cmaptools, disponible en: [cmap.ihmc.us/](http://cmap.ihmc.us/)

Figura3. Ejemplos de salidas desde SPARQL Endpoint<sup>5</sup>.

<b>Tabla Resumen del Vocabulario de Títulos</b>		
<b>Campos</b>	<b>Predicados RDF</b>	<b>Tipo de RDF</b>
Nombres	FOAF:NAME	Property
Cedula	DC:IDENTIFIER	Property
Nacionalidad	EDM:COUNTRY	Property
Registro	DC:ID	Property
Título	FOAF:TITLE	Property
Provincia	FOAF:BASED-NEAR	Property
Title	RDFS:LABEL	Property
Name	RDFS:LABEL, SKOS:PRELABEL	Property
Description	SKOS:DEFINITION	Property
Homepage	FOAF:PAGE	rel
Username	FOAF:NAME	Property
Parent	SKOS:BROADER	rel
Changed	DC:MODIFIED	property
Uid	SIOC:HAS-CREATOR	rel
Created	DC:CREATED	property
Título	-	SIOC:ITEM, FOAF:DOCUMENT
Vocabulario	-	SKOS:COLLECTION
Termino	-	SKOS:CONCEPT

Tabla2. Tabla resumen del Vocabulario de la Implementación.

<sup>5</sup> Ejemplos sacados del SPARQL Endpoint disponible en: <http://syberiaec.com>

# CAPITULO 4. Visualización de Información

---

## Visualizando metadatos

### 4.1. Visualización de Metadatos

La visualización de metadatos ayuda a proveer una mirada global del contenido de los metadatos que se encuentran en un repositorio RDF y permite comparar muchos ítems o describir detalladamente los ítems en forma individual, esta visualización de metadatos puede utilizar técnicas tradicionales para el efecto como: las de recuperación de información, las de visualización tradicional y las de mining visual de datos. Los metadatos pueden ser visualizados de forma individual en salidas como tablas, grillas, listas, etc. O salir en forma de gráficos o inclusive imágenes que muestren resúmenes de los metadatos o agrupaciones de los mismos, como sustenta Herskovic Valeria en [18].

Para las dos formas globales de visualizar los datos existen herramientas, Plugin, APIS y módulos que nos brindan facilidad para su creación y configuración, y muchas de ellas gratuitas. La visualización de metadatos puede abarcar temas tan diversos como una tabla, grillas, listas, gráficas y hasta metadatos en imágenes, con sus propios vocabularios y propiedades relacionadas, como sustenta Herskovic Valeria en [18].

El escogimiento de las herramientas de visualización más idóneas requiere de un análisis de las mismas.

### 4.1. Visualización de metadatos con SPARQL

Las visualizaciones de metadatos pueden utilizar la ayuda de los lenguajes de consulta como SPARQL, que produzcan salidas de datos requeridos, o pueden ser producto de consultas de lenguajes comunes como SQL. En cuanto a las consultas del contenido que puede ser localizado mediante URIS por lenguajes como SPARQL, sus salidas pueden mostrarse en algunos formatos, pero existen muy pocas herramientas para visualizar los datos extraídos al alcance del usuario común, debido al poco desarrollo en este ámbito. En el caso de DRUPAL existen módulos y herramientas para visualizar datos y hacer gráficas. Pero están enfocadas mayormente a datos normales, el módulo más representativo en cuanto a visualizaciones semánticas en DRUPAL es SPARQL\_VIEWS, que nos provee la posibilidad de juntar nuestras consultas y colocarlas en vistas, aunque aún sean visualizaciones sencillas y con muchos bugs por resolver.

## **4.2. Plugins o herramientas para generar gráficos**

Existen muchas herramientas en muchos lenguajes de programación que permiten generar gráficos de datos con diferentes características cada una de ellas y con diferente dificultad de configuración, algunas de estas herramientas han generado módulos listos para DRUPAL que permiten conectar los datos y bibliotecas externas e internas para generar vistas de gráficos de los nodos, el contenido o de vistas SPARQL de contenido propio o externo.

A continuación un sucinto análisis de algunas herramientas para gráficos actuales:

### **4.2.1. Google ChartAPI**

Este módulo permite hacer mapas y charts especiales de google, trabaja con el módulo de vistas de DRUPAL y con el módulo de vistas SPARQL. Además este API permite realizar distintos tipos de gráficos, desde barras hasta gráficos de pastel en 3D. El módulo es gratuito, además de ser ligero y no necesitar de ninguna librería del lado del servidor, se integra directamente como módulo a DRUPAL y no consume demasiada memoria, pero está limitado diariamente por google a 50.000 peticiones por día.

#### **4.2.1.1. Tipos de Gráficas**

Entre los tipos de gráficos que google chart permite realizar están: gráficos de barra, línea, pastel, radar, google o meter y otras variantes más de este tipo. Sin embargo para visualizaciones de datos semánticos, el módulo de google chart para DRUPAL tiene muchos vacíos e inconvenientes a la hora de crear las gráficas.

#### **4.2.1.2. Personalización**

Google chart API permite personalizar aspectos visuales y de contenido como: el tamaño del gráfico, etiquetas y títulos. Y en gráficas que permiten cargas más fuertes de datos se permite la configuración de varios juegos de datos, como refiere Álvarez Ángel en [19].

Por las ventajas en compatibilidad, accesibilidad y variedad del módulo se lo escogió para la visualización de datos RDF y registros del SENESCYT. Algunos ejemplos de gráficas con google chart API.



Figura4. Imágenes representativas deGoogle chart API<sup>6</sup>.

#### 4.2.2. PREFUSE

PREFUSE está hecho en lenguaje JAVA y es una herramienta que sirve para crear gráficos de información serializable, que soporta ACTION SCRIPT. PREFUSE ofrece una gran variedad de configuraciones de los gráficos para setear los valores correspondientes, este kit de herramientas posee compatibilidad con un gran número de funciones, para modelar los datos y presentarlos en diversas formas, y su desarrollo es abierto a la comunidad de desarrolladores web.

##### 4.2.2.1. Tipos de Gráficas

Entre los tipos de gráficos que PREFUSE permite realizar gracias están: gráficos de barra, línea, árboles y gráficos de incertidumbre, arboles, plano de flujo, gráficos radiales, tipo eyefish, y otros más. Este toolkit utiliza librería de JAVA con integración a aplicaciones del tipo Swing o Applets.

<sup>6</sup> Recopilación información de: [code.google.com/intl/es/APIS/chart/](http://code.google.com/intl/es/APIS/chart/)

#### 4.2.2.2. Personalización

PREFUSE puede ser configurado permitiendo personalizar aspectos visuales y de contenido como: el tamaño del gráfico, etiquetas y títulos y en gráficas que permiten cargas más fuertes de datos es permitida la configuración de varios juegos de datos, como refiere GITHUB en [22].

Esta herramienta no es parte de los módulos de DRUPAL pero podría ser implementada en aplicaciones hechas en JAVA o que soporten librería JAVA, esta potente librería no ha sido ocupada en la presente implementación por la falta de compatibilidad y porque se ha usado otras librerías en JAVASCRIPT similares.

Algunos ejemplos de gráficas con PREFUSE.

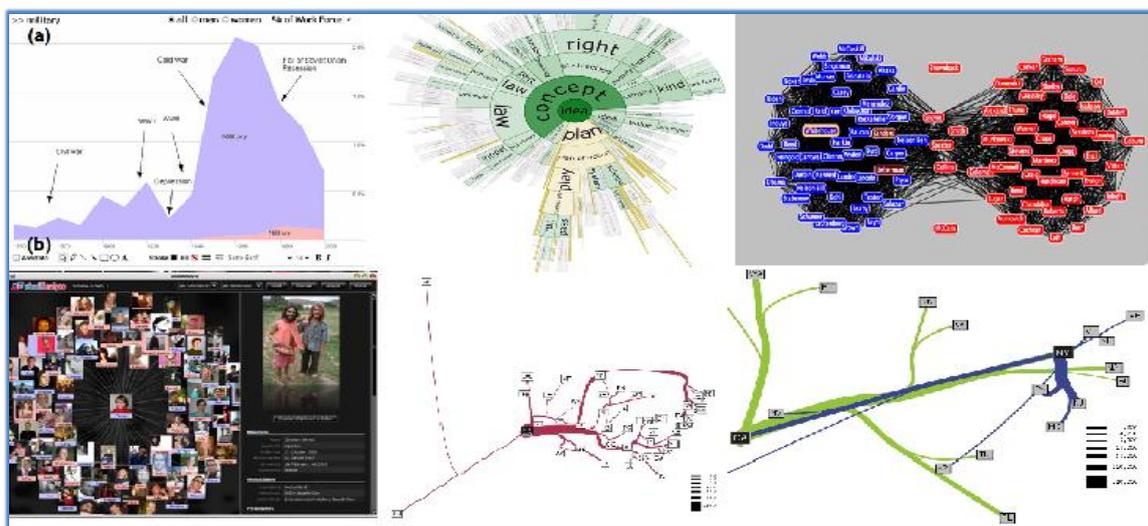


Figura5. Imágenes representativas de PREFUSE<sup>7</sup>.

#### 4.2.3. D3

D3.js es una librería hecha en JAVASCRIPT que permite enlazar datos al DOM de la página y formar gráficos, tablas o listas con ellos. D3 manipula eficientemente los documentos sobre una base de datos establecidos, integrándose completamente con Css3, HTML y SVG disminuyendo coste de tiempo en representaciones propias. Es muy rápido y puede trabajar con grandes cantidades de datos, además de permitir reutilización de código a través de módulos. D3 reduce el código empleado para la visualización de datos con respecto a otros como DOM API, además de ser compatible con todos los navegadores actuales. Esta librería es la versión mejorada de otra librería utilizada para gráficos llamada Protovis, que permitía hacer visualizaciones de marcos estáticos y por eso su desarrollador pensó en rearmar toda esta librería y hacerla dinámica.

<sup>7</sup> Recopilación información de: [PREFUSE.org/](http://PREFUSE.org/)

#### 4.2.3.1. Tipos de Gráficas

D3 cuenta con un buen número de tipos de gráficos empezando por tablas, listas, gráficos tipo diagrama, barras apiladas, diagrama de dispersión, pirámide poblacional, mapas, calendario, Árbol de nodos, gráficos de burbujas, círculo de embalaje y diagramas de caja.

#### 4.2.3.2. Personalización

D3 puede especificar las funciones y atributos en función del dato almacenado y no solo en simples constantes. Posee un amplio número de funciones preconcebidas que pueden ser reutilizadas en función de la necesidad, cuenta con características para el enlazado de datos a selecciones definidas pues los puede manejar como matrices de valores. Se puede manejar datos para gráficas para que no solo se muestren de manera estática, sino que se vayan actualizando dinámicamente, como refiere github en [23].

Aunque esta librería es muy completa se ha preferido trabajar con otras por mayor compatibilidad, además de tener otros módulos listos para trabajar con DRUPAL.

Algunos ejemplos de gráficas con D3.

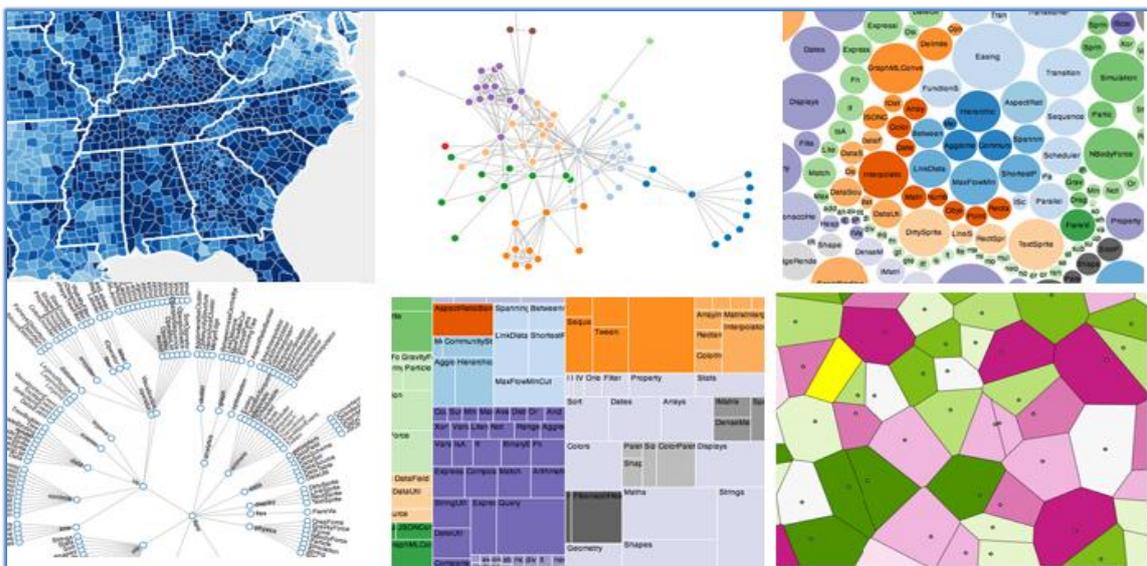


Figura6. Imágenes representativas deD3<sup>8</sup>.

#### 4.2.4. RAPHAEL

Raphaël Es una librería hecha en JAVASCRIPT que se utiliza en visualización de datos. Raphaël utiliza estándares de la W3C para las funciones que maneja en la creación de gráficas para que los gráficos creados sean parte del DOM del documento y se pueda ubicar y trabajar con estas partes por separado. Y agregar controladores de eventos de JAVASCRIPT y AJAX, y posee compatibilidad con los exploradores actuales. Raphaël cuenta ya con su propio

<sup>8</sup> Recopilación información de: [mbostock.github.com/d3/](http://mbostock.github.com/d3/)

módulo para DRUPAL y su librería se puede usar en otros módulos por separado ayudando a proveer las características de Raphaël.

#### 4.2.4.1. Tipos de Gráficas

Raphaël provee soporte para varios tipos de gráficos como pastel, mapas, diagramas, gráficas analíticas, burbujas, su librería es muy versátil y la ocupan otras herramientas de visualización para gráficas como Morris, un módulo de DRUPALVIEWS\_SLIDESHOW que muestra gráficas tipo slideshows de los datos.

#### 4.2.4.2. Personalización

Raphaël permite trabajar con eventos del DOM pues los datos con los que trabaja son tomados como objetos que pueden ser referenciados en funciones logrando hacer dinámicos los gráficos. Raphaël cuenta con un nutrido grupo de funciones propias para el diseño y también para la funcionalidad del gráfico, también posee un potente conjunto de fórmulas con las que se puede trabajar, como sustenta Baranovskiy Dimitrien [24].

Esta librería también ha sido utilizada para las visualizaciones, específicamente en la vista “título” que saca los resultados en forma de slideshows.

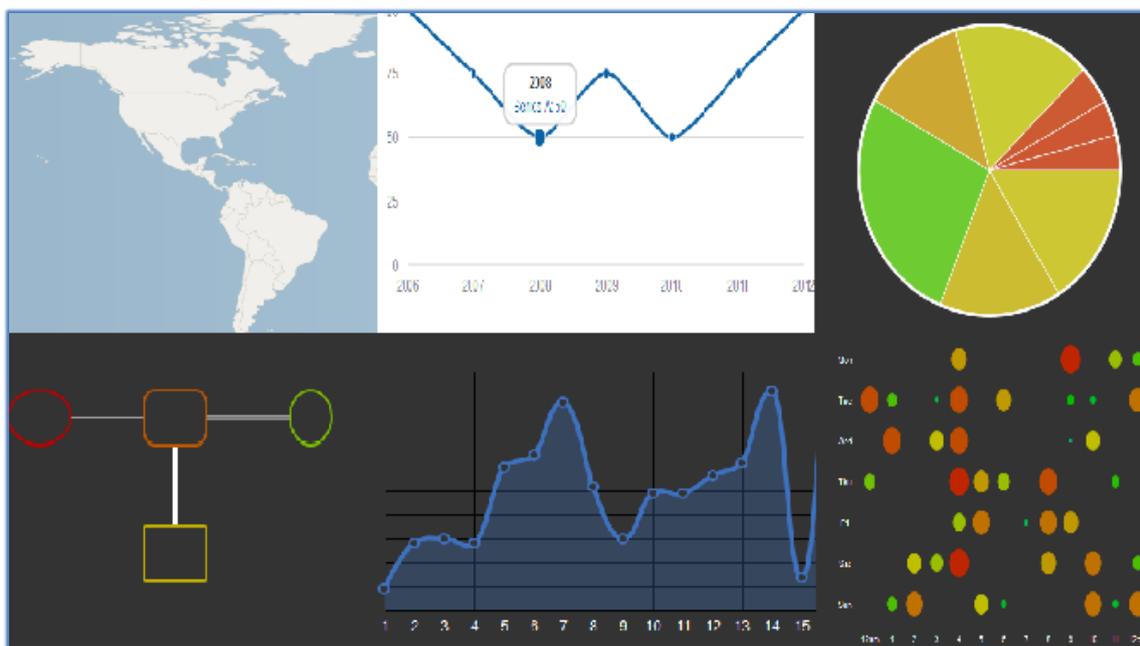


Figura7. Imágenes representativas deRaphaël<sup>9</sup>.

En las etapas de descripción y dominio del problema se ha descrito el problema de la web actual y como esta enfocada a la publicación de documentos y el hipertexto, que nos brindan referencias a más información, como en el caso de los datos de los registros de la Senescyt

<sup>9</sup> Recopilación información de: [RAPHAELjs.com/](http://RAPHAELjs.com/)

que se utilizan para el proyecto y como la WEB semántica puede lograr ser una solución a este problema brindando características necesarias para que los datos allí expuestos sean reutilizables y fácilmente entendibles por maquinas de gestión de información.

En la etapa de adquisición de datos, procesamiento y codificación en RDF se describió como estos datos han sido adquiridos, como posteriormente se los ha manipulado para que cumplan con términos de estructura del vocabulario y como se los ha introducido en las tablas de la base de datos del repositorio RDF, para que se transformen en metadatos.

### 4.3. *Visualización de metadatos del SENESCYT*

Para la visualización de los metadatos del SENESCYT, se ha escogido la herramienta de Google "Google Chart API", que a través de un módulo hecho para DRUPAL, nos facilita la tarea de configuración de los gráficos, además de que trabaja 100% con el módulo de vistas y de vistas SPARQL de DRUPAL, que son parte de los módulos con los que se ha trabajado para la implementación del proyecto, además de utilizar la librería Raphaël.js con la cual se generaron graficas estadísticas con resultados obtenidos de los metadatos.

Estas visualizaciones no pretenden abarcar un espectro estadístico completo, más bien son ejemplos de cómo se puede visualizar los metadatos semánticos que estén en un repositorio, como también en Repositorios externos. Las visualizaciones del proyecto no son demasiado complejas, pero son una clara muestra de lo que podemos visualizar con los metadatos. Además de las librerías para generar gráficos hemos trabajado con el módulo VIEWS y SPARQL\_VIEWS que nos ayudan a la configuración de las salidas. SPARQL\_VIEWS nos brinda el soporte de consulta para sacar las vistas necesarias. El módulo SPARQL realiza muchas acciones al efectuar una vista separando las etiquetas y nombres de recursos solicitados por la vista, a nivel de sentencias SPARQL se efectúan algunos procesos veamos uno de los importantes a nivel de código SPARQL.

```
// Attach all the bundles that have been defined by the user.
$dbundles = db_query("SELECT name, label FROM {sparql_views_resource_type}")->fetchAll();
foreach ($bundles as $key => $bundle) {
  $items['sparql_views_resource']['bundles'][$bundle->name] = array(
    'label' => $bundle->label,
    'admin' => array(
      'path' => 'admin/structure/sparql_views/manage/%sparql_views_resource_type',
      'real path' => 'admin/structure/SPARQL-VIEWS/manage/' . $bundle->name,
      'bundle argument' => 4,
      'access arguments' => array('administer sparql_views types'),
    ),
  );
}
```

Se justifica el uso de este módulo pues nos da la posibilidad de interactuar con RDF y SPARQL endpoint y mostrarlo en una vista proporcionando toda la ayuda necesaria para el formato de salida.

### 4.3.1.1. Vista “Registro de títulos”.

Esta vista muestra el contenido de la página del tipo “títulos” que fue mapeado a RDF siendo mostrado como una Tabla.

Cedula	Nacionalidad	Registro	Título	Nombres y Apellidos	Nodo
1714492368	Ecuatoriana	100-300-799-900	Ingeniero en Informática y Sistemas Computacionales	crystian antonio haro vinueza	1
0401881808	Ecuatoriana	100-345-890-101	Ingeniero en Finanzas	Nelly Azucena Morales Morales	2
0401881809	Ecuatoriana	100-345-890-102	Ingeniero en Telecomunicaciones	Suntaxi Cocanguilla Jairo Medardo	3
908043557	Ecuatoriana	1006-05-570036	ADMINISTRADOR EDUCATIVO	ACEVEDO AYALA RICARDO ALFREDO	7
1801091305	Ecuatoriana	1010-02-141332	LICENCIADO CIENCIAS DE LA EDUCACION	ACOSTA A ESTHELA INES	8
1801091305	Ecuatoriana	1010-02-141332	LICENCIADO CIENCIAS DE LA EDUCACION	ACOSTA A ESTHELA INES	9
913342382	Ecuatoriana	1006-07-767471	LICENCIADA EN CIENCIAS DE LA EDUCACION ESPECIALIZACION HISTORIA Y GEOGRAFIA	ACOSTA ACOSTA AMARILIS NARCISA	10
913342382	Ecuatoriana	1006-07-767471	LICENCIADA EN CIENCIAS DE LA EDUCACION ESPECIALIZACION HISTORIA Y GEOGRAFIA	ACOSTA ACOSTA AMARILIS NARCISA	11
909119299	Ecuatoriana	1039-03-460095	INGENIERO MECANICO	ACOSTA ACOSTA ANGEL LEONARDO	12
909119299	Ecuatoriana	1039-03-460095	INGENIERO MECANICO	ACOSTA ACOSTA ANGEL LEONARDO	13
1703010494	Ecuatoriana	1005-08-852021	LICENCIADA EN CIENCIAS DE LA EDUCACION PROFESORA DE ENSEÑANZA SECUNDARIA EN LA ESPECIALIZACION DE INGLES	ACOSTA ACOSTA BEATRIZ MARIA EUGENIA	14

Figura8. Vista “registro de títulos”.

QUERYSPARQL para esta vista:

```

PREFIX FOAF: <http://XMLns.com/FOAF/0.1/>
PREFIX DC: <http://purl.org/DC/elements/1.1/>
PREFIX DC: <http://purl.org/DC/terms/>
PREFIX RDFS: <http://www.w3.org/2000/01/RDF-schema#>
PREFIX SIOC: <http://RDFS.org/SIOC/ns#>
PREFIX SKOS: <http://www.w3.org/2004/02/SKOS/core# >
PREFIX RDF: <http://www.w3.org/1999/02/22-RDF-syntax-ns#>
PREFIX site: <http://www.syberiaec.com/ns#>
SELECT ?name ?title ?identifier ?country ?based_near ?id
WHERE {
  ?nombres foaf:name ?name.
  ?titulo foaf:title ?title.
  ?cedula foaf:identifier ?identifier.
  ?nacionalidad dc:country ?country.
  ?Provincia foaf:based_near ?based_near.
  ?Registro dc:id ?id.}
LIMIT 16

```

### 4.3.1.2. Vista “Totales por Provincia”.

Esta es una vista SPARQL view que agrupa los diferentes títulos por provinciaponiendo sus totales en una lista ordenada y luego de cada provincia muestra los diferentes títulos con los números de registros asociados a ellos en una gráfica tipo Pie 3d.

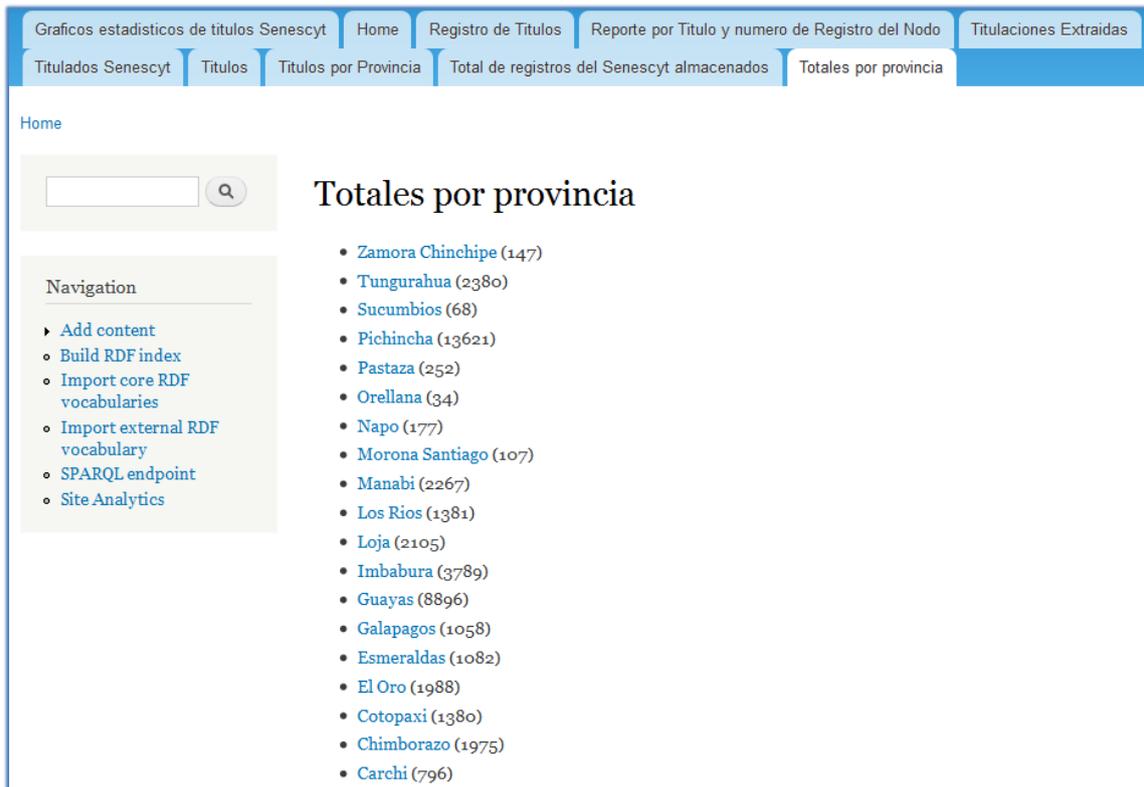


Figura9. Vista “Título y registro del nodo”.

QUERYSPARQL para esta vista:

```
PREFIX FOAF: <http://XMLns.com/FOAF/0.1/>
PREFIX DC: <http://purl.org/DC/elements/1.1/>
PREFIX DC: <http://purl.org/DC/terms/>
PREFIX RDFS: <http://www.w3.org/2000/01/RDF-schema#>
PREFIX SIOC: <http://RDFS.org/SIOC/ns#>
PREFIX SKOS: <http://www.w3.org/2004/02/SKOS/core# >
PREFIX RDF: <http://www.w3.org/1999/02/22-RDF-syntax-ns#>
PREFIX site: <http://www.syberiaec.com/ns#>
```

```
SELECT ?based_near ?title
WHERE
{
  ?titulofoaf:based_near ?based_near.
  ?Registrodc:id ?id.
}
Group by ?based_near
```

### 4.3.1.3. Vista “Titulados SENESCYT”.

Esta es una vista SPARQL\_ view que extrae las etiquetas RDF de cédula, nombres, nacionalidad, registro y título mediante una consulta SPARQL y los muestra en la vista a través de una grilla con links a los nodos correspondientes.

The screenshot shows the 'Titulados Senescyt' web application. The header includes the logo and title 'Principios Semánticos aplicados a registros del Senescyt' with a subtitle 'Implementación de Proyecto de tesis UTPL Crystian Haro Yñueza 2011'. The navigation bar contains links for Home, Registro de Titulos, Reporte por Titulo y numero de Registro del Nodo, Titulaciones Extraidas, Titulados Senescyt (selected), and Titulos por Provincia. Below the navigation bar, there is a search bar and a 'Items per page' dropdown set to 16. The main content area displays a grid of 8 record cards, each containing the following information:

Registro	Registro	Registro	Registro
Nombres: ACEVEDO AYALA RICARDO ALFREDO Cedula: 908043557 Nacionalidad: Ecuatoriana Registro: 1006-05-570036 Titulo: ADMINISTRADOR EDUCATIVO URI: <a href="http://syberiaec.com/node/7">http://syberiaec.com/node/7</a> Registro No: 1	Nombres: ACOSTA A ESTHELA INES Cedula: 1801091305 Nacionalidad: Ecuatoriana Registro: 1010-02-141332 Titulo: LICENCIADO CIENCIAS DE LA EDUCACION URI: <a href="http://syberiaec.com/node/8">http://syberiaec.com/node/8</a> Registro No: 2	Nombres: ACOSTA A ESTHELA INES Cedula: 1801091305 Nacionalidad: Ecuatoriana Registro: 1010-02-141332 Titulo: LICENCIADO CIENCIAS DE LA EDUCACION URI: <a href="http://syberiaec.com/node/9">http://syberiaec.com/node/9</a> Registro No: 3	Nombres: ACOSTA ACOSTA AMARILIS NARCISA Cedula: 913342382 Nacionalidad: Ecuatoriana Registro: 1005-07-767471 Titulo: LICENCIADA EN CIENCIAS DE LA EDUCACION ESPECIALIZACION HISTORIA Y GEOGRAFIA URI: <a href="http://syberiaec.com/node/11">http://syberiaec.com/node/11</a> Registro No: 4
Nombres: ACOSTA ACOSTA AMARILIS NARCISA Cedula: 913342382 Nacionalidad: Ecuatoriana Registro: 1005-07-767471 Titulo: LICENCIADA EN CIENCIAS DE LA EDUCACION ESPECIALIZACION HISTORIA Y GEOGRAFIA URI: <a href="http://syberiaec.com/node/10">http://syberiaec.com/node/10</a>	Nombres: ACOSTA ACOSTA ANGEL LEONARDO Cedula: 909119299 Nacionalidad: Ecuatoriana Registro: 1039-03-460095 Titulo: INGENIERO MECANICO URI: <a href="http://syberiaec.com/node/13">http://syberiaec.com/node/13</a>	Nombres: ACOSTA ACOSTA ANGEL LEONARDO Cedula: 909119299 Nacionalidad: Ecuatoriana Registro: 1039-03-460095 Titulo: INGENIERO MECANICO URI: <a href="http://syberiaec.com/node/12">http://syberiaec.com/node/12</a>	Nombres: ACOSTA ACOSTA BEATRIZ MARIA EUGENIA Cedula: 1703010494 Nacionalidad: Ecuatoriana Registro: 1005-08-852021 Titulo: LICENCIADA EN CIENCIAS DE LA EDUCACION PROFESORA DE ENSEÑANZA SECUNDARIA EN LA ESPECIALIZACION DE INGLES

Figura10. Vista “Titulados SENESCYT”.

QUERYSPARQL para esta vista:

```
PREFIX FOAF: <http://XMLns.com/FOAF/0.1/>
PREFIX DC: <http://purl.org/DC/elements/1.1/>
PREFIX DC: <http://purl.org/DC/terms/>
PREFIX RDFS: <http://www.w3.org/2000/01/RDF-schema#>
PREFIX SIOC: <http://RDFS.org/SIOC/ns#>
PREFIX SKOS: <http://www.w3.org/2004/02/SKOS/core#>
PREFIX RDF: <http://www.w3.org/1999/02/22-RDF-syntax-ns#>
PREFIX site: <http://www.syberiaec.com/ns#>
SELECT ?name ?title ?account ?country?based_near ?license?resourceURI
WHERE {
?nombres foaf:name ?name.
?títulofoaf:title ?title.
?cédula dc:identifier?identifier.
?nacionalidadog:country-name ?country-name.
?ProvinciaFOAF:based_near ?based_near.
?RegistroDC:ID ?id.
?resourceURI<http://syberiaec.com/SPARQL>
```

LIMIT@

#### 4.3.1.4. Vista “Títulos por Provincia”.

Esta es una vista SPARQL\_ VIEWS que extrae las etiquetas RDF de Título, nombres mediante una consulta SPARQL y las muestrasen la vista a través de una Tabla añadiendo tres filtros de Búsqueda uno para búsquedas por Provincia, otro por Título y otro por nombre del titulado.

Principios Semánticos aplicados a registros del Senescyt  
Implementación de Proyecto de tesis UTPL Crystian Haro Vinuesa 2011

Home Registro de Títulos Reporte por Título y numero de Registro del Nodo Titulaciones Extraidas Titulados Senescyt

Titulos por Provincia Total de registros del Senescyt almacenados

Home

Navigation

- ▶ Add content
- Build RDF index
- Import core RDF vocabularies
- Import external RDF vocabulary
- SPARQL endpoint

Titulos por Provincia

Busqueda por Provincia: Loja

Busqueda por Título

Busqueda por Nombres: Lopez Vargas

Apply

Provincia: Loja

Titulo: INGENIERO QUIMICO Nombres y Apellidos: LOPEZ VARGAS JAVIER ALFREDO Total de Títulos: 1	Titulo: INGENIERO EN SISTEMAS INFORMATICOS Y COMPUTACION Nombres y Apellidos: LOPEZ VARGAS JORGE AFRANIO Total de Títulos: 2	Titulo: LICENCIADA EN SECRETARIADO EJECUTIVO Nombres y Apellidos: LOPEZ VARGAS MARIA VERONICA Total de Títulos: 3
--	--	---

Figura12. Vista de Títulos por Provincia.

QUERYSPARQL para esta vista:

```
PREFIX FOAF: <http://XMLns.com/FOAF/0.1/>
PREFIX DC: <http://purl.org/DC/elements/1.1/>
PREFIX DC: <http://purl.org/DC/terms/>
PREFIX RDFS: <http://www.w3.org/2000/01/RDF-schema#>
PREFIX SIOC: <http://RDFS.org/SIOC/ns#>
PREFIX SKOS: <http://www.w3.org/2004/02/SKOS/core#>
PREFIX RDF: <http://www.w3.org/1999/02/22-RDF-syntax-ns#>
PREFIX site: <http://www.syberiaec.com/ns#>
SELECT ?name ?title?country-name ?based_near ?id
WHERE { ?nombres FOAF:NAME ?name.
?tituloFOAF:TITLE ?title.
?nacionalidadog:country-name ?country-name.
?ProvinciaFOAF:based_near ?based_near.
?RegistroDC:ID ?id
FILTER(?based_near). FILTER(?title). FILTER(?name).}
LIMIT3
```

#### 4.3.1.5. Vista “Títulos”.

Esta es una vista SPARQL\_ VIEWS que extrae las etiquetas RDF de Título, nombres mediante una consulta SPARQL y las muestras en la vista a través de SlideShows agrupándolos por Título y luego por provincia para poder ver el número de títulos en cada provincia. Añadiendo un filtro de Búsqueda tipo Jump-Menu que nos permite escoger el título a buscar.

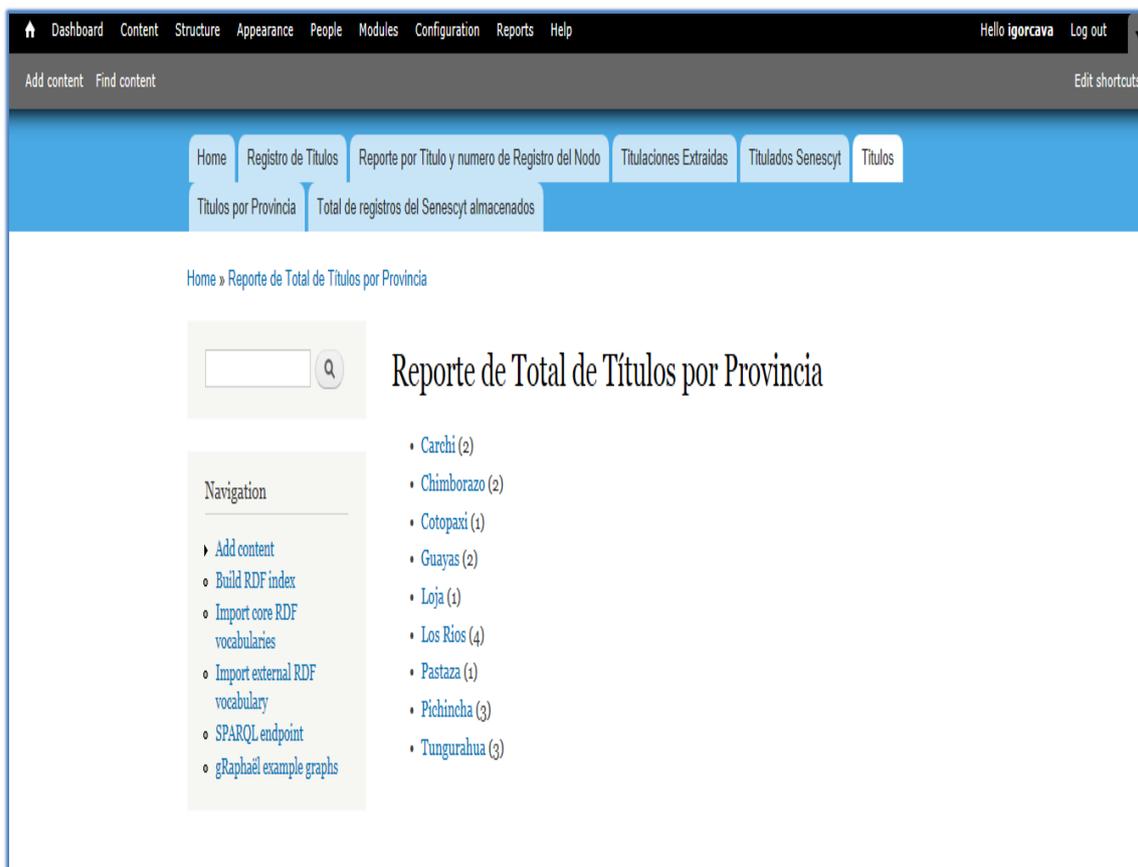


Figura13. Vista de Títulos.

QUERYS PARQL para esta vista:

```
PREFIX FOAF: <http://XMLns.com/FOAF/0.1/>
PREFIX DC: <http://purl.org/DC/elements/1.1/>
PREFIX DC: <http://purl.org/DC/terms/>
PREFIX RDFS: <http://www.w3.org/2000/01/RDF-schema#>
PREFIX SIOC: <http://RDFS.org/SIOC/ns#>
PREFIX SKOS: <http://www.w3.org/2004/02/SKOS/core# >
PREFIX RDF: <http://www.w3.org/1999/02/22-RDF-syntax-ns#>
PREFIX site: <http://www.syberiaec.com/ns#>

SELECT ?title ?id
WHERE
{
  ?tituloFOAF:TITLE ?title.
  ?RegistroDC:ID ?id.
}
```

Group by ?title  
FILTER(?title).  
LIMIT 10

#### 4.3.1.6. Grafico “Total de Títulos por Provincia”.

Este es un gráfico generado con ayuda de la biblioteca RAPHAEL que muestra el total de títulos por cada provincia en un gráfico de tipo Barra vertical.

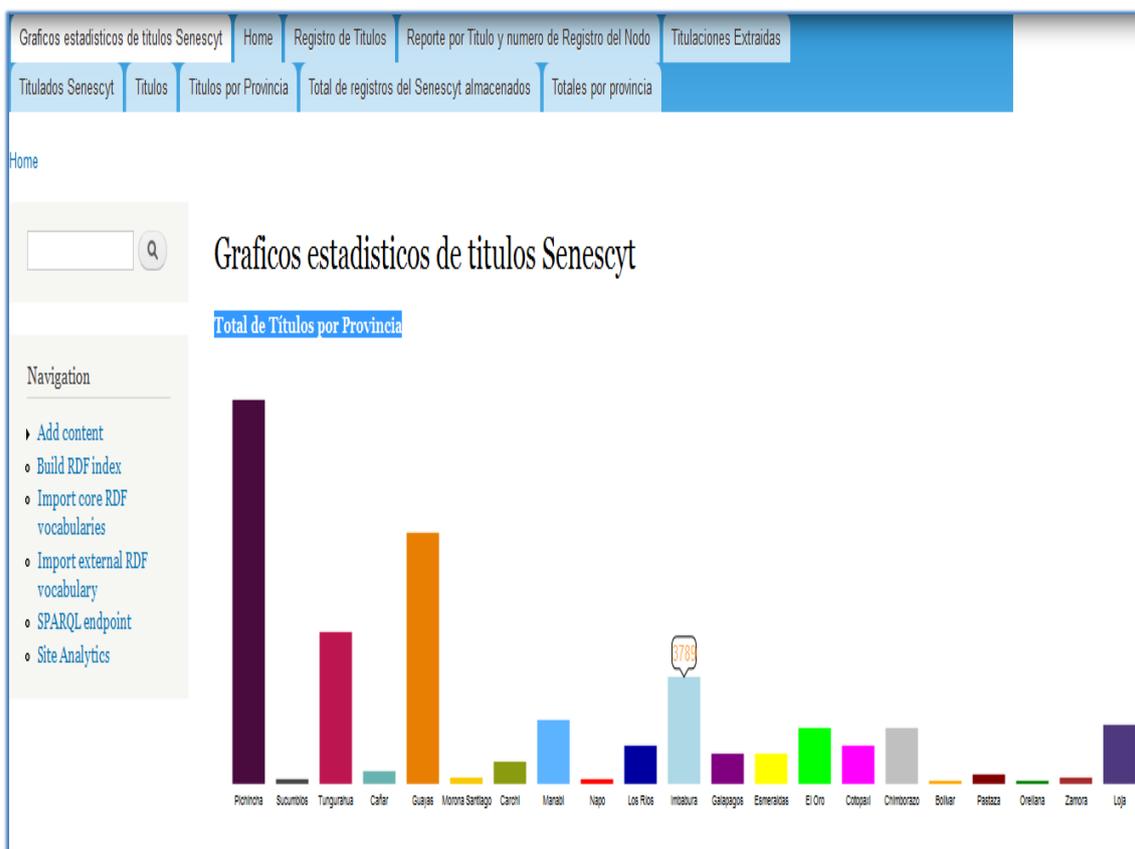


Figura14. Grafico Total de Títulos por Provincia.

#### 4.3.1.7. Grafico “Total de Títulos por Provincia en barra horizontal”.

Este es un gráfico generado con ayuda de la biblioteca RAPHAEL que muestra el total de títulos por cada provincia en un gráfico de tipo Barra horizontal.



Figura15. Grafico Total de Títulos por Provincia hbar.

#### 4.3.1.8. Grafico “Porcentaje de Títulos por Provincia”.

Este es un gráfico generado con ayuda de la biblioteca RAPHAEL que muestra los porcentajes de las provincias con mayor número de títulos en un gráfico de tipo Pie.

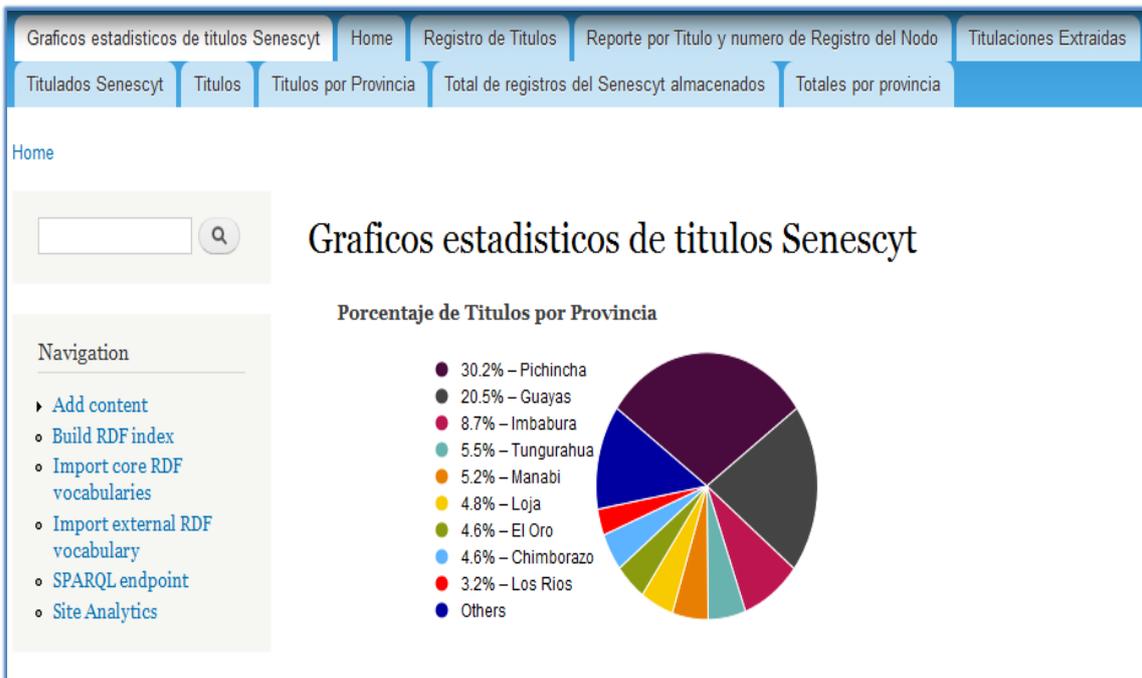


Figura16. Grafico Porcentaje de Títulos por Provincia.

#### 4.3.1.9. Grafico “Porcentaje de Profesionales por Título”.

Este es un gráfico generado con ayuda de la biblioteca RAPHAEL que muestra los porcentajes de los Títulos con mayor número de titulados en un gráfico de tipo Pie.



Figura17. Grafico Porcentaje de Profesionales por Título.

#### 4.3.1.10. Grafico “Top 25 de las carreras más seguidas.”

Este es un gráfico generado con ayuda de la biblioteca RAPHAEL que muestra los porcentajes de las provincias con mayor número de títulos en un gráfico de tipo Pie.



**Figura18. Grafico Porcentaje de Profesionales por Título.**

# CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES

---

## De la mano con la semántica

### 6.1. Conclusiones Generales.

- ✓ El uso de vocabularios pre establecidos y estandarizados representan un gran soporte en la generación de nuevos proyectos enfocados a soluciones semánticas, mejorando la accesibilidad en la nube semántica. La flexibilidad con la que se pueden utilizar las propiedades y clases de los distintos Vocabularios Ontológicos de libre acceso, para la creación de nuevos y personalizados Vocabularios, logra representar y compartir el conocimiento mediante el uso de una estructura estandarizada.
- ✓ El uso de principios semánticos, ayuda a transformar datos simples en estructuras bien organizadas que aportan conocimiento a la información.
- ✓ Los repositorios RDF aportan conocimiento a los datos normales almacenados en una base de datos relacional, porque a través de ellos se logra obtener la información contenida en forma de tripletas. Ya que gracias a su sencillo modelo de datos es posible manejar datos más complejos a menor costo.
- ✓ Los datos consultados vía SPARQL y devueltos en formato RDF, TURTLE, XML u otros, son el medio de obtención de los metadatos almacenados en el repositorio RDF, cuyos resultados se pueden usar para generar gráficos permitiendo una mejor visualización de los resultados.

## **6.2. Recomendaciones.**

- ✓ Se podría construir aplicaciones semánticas que puedan utilizar lenguajes de consulta tradicionales como SQL, que logren obtener metadatos con las mismas sentencias tradicionales.
- ✓ Se recomienda implementar aplicaciones semánticas siempre pensando en la utilidad y practicidad, más que en el tamaño de las mismas, implementaciones que ayuden a incrementar la nube semántica y poner nuevos datos en la misma.
- ✓ El uso de vocabularios estándar, es una buena práctica para este tipo de aplicaciones, pues disminuye el tiempo de desarrollo, ya que en la creación de nuevos vocabularios se tiene que considerar muchos parámetros y normas que se deben seguir.
- ✓ La utilización de aplicaciones CMS como DRUPAL para implementar principios semánticos, son de gran ayuda por su robustez y enfoque hacia este tipo de tecnologías, además cuentan con soporte de una gran comunidad de colaboradores.
- ✓ Se podría desarrollar aplicaciones semánticas más robustas con lenguajes como ORACLE, que cuenta con una infraestructura técnica y humana que permiten obtener un soporte de calidad.

### 6.3. Casos de uso.

<b>Nombre:</b>	Ingresos de nuevo contenido tipo Título
<b>Descripción:</b>	El usuario añade un nuevo contenido tipo Título.
<b>Actor:</b>	Usuario
<b>Precondiciones:</b>	El usuario debe tener permiso para añadir nuevo contenido.
<b>Flujo Normal:</b>	
<b>Usuario</b>	<b>Aplicación DRUPAL</b>
1.-Navega a la Página de Registros <a href="http://syberiaec.com">http://syberiaec.com</a>	
2.-Se logea y añade nuevo contenido de tipo TÍTULO llenando los campos correspondientes y guarda.	
	3.-Se receiptael valor del formulario y la aplicación sigue una serie de subprocesos para poder emitir el resultado.
	4.-Se presenta en pantalla el resultado de la adición.
<b>EXTENSIONES:</b>	
3ª.-Los valores ingresados no deben ser repetidos.	

<b>Nombre:</b>	Crear nueva vista
<b>Descripción:</b>	El usuario crea una nueva vista externa o interna
<b>Actor:</b>	Aplicación DRUPAL
<b>Precondiciones:</b>	Debe tener permiso para poder añadir vistas.
<b>Flujo Normal:</b>	
<b>Aplicación DRUPAL</b>	<b>Repositorio DRUPAL</b>
1.-el usuario añade una nueva vista el nombre y escoge su tipo y formato guarda.	
	2.-se crea la vista y se guarda en cache y el sistema muestra la página de la vista en si para que el usuario pueda editarla.
4.-El usuario edita la vista con los parámetros a su elección y la guarda	
	5.-El sistema la guarda y presenta en pantalla el resultado de la vista.
<b>EXTENSIONES:</b>	
3ª.-El nombre de la vista no debe existir.	
4.-el nombre del path tampoco debe existir.	

# Anexo

---

## Anexo A

### Instalación de OUTWITHUB

#### Requerimientos

- 1.0- Es necesario instalar el navegador MOZZILLAFIREFOX.
- 2.0- SI OUTWIT necesita un Plugin adicional, FIREFOX se encargará de descargarlo e instalarlo automáticamente.

#### Instalación

- 1.0- Se ejecuta FIREFOX y se procede a la descarga del Plugin OUTWIT desde la sección descargas <http://www.OUTWIT.com/products/hub/>
- 2.0- Una vez descargado el complemento se instala y reinicia FIREFOX, la extensión se anclará a la barra de herramientas para un más rápido acceso.
- 3.0- Se abre la herramienta para comenzar a utilizarla seteadando los valores para extraer los registros requeridos desde la página de la SENESCYT.
- 4.0- Una vez que se abre la ventana de hub Wit lo primero que se tiene que hacer es entrar a la opción "page" aquí se le puede decir a OUTWIT la página de la que se extraerán los datos. Para este proyecto se ingresó la página de títulos académicos de la SENESCYT.
- 5.0- Luego de que se despliegue la página se escoge la opción de búsqueda por apellidos y se puede comenzar a ingresar apellidos. Por medio de guía telefónica de Quito se ingresa se toma los apellidos para la muestra.
- 6.0- Se prueba con un apellido de la letra A por ejemplo: Acurio con la opción buscar, se desplegarán los diferentes nombres para ese apellido, luego se escoge la opción links para verificar los links que dirijan hacia los datos individuales.
- 7.0- Para la salida de los datos se debe setear la opción de scrapper que tiene Out Wit que permite dar formato a los datos de salida conforme a los datos encontrados y también escoger la posición y título de las columnas en el dataset de salida que tendrá la consulta, además de poder decirle a Out Wit entre que etiquetas o palabras se encuentra los datos como muestra la figura.

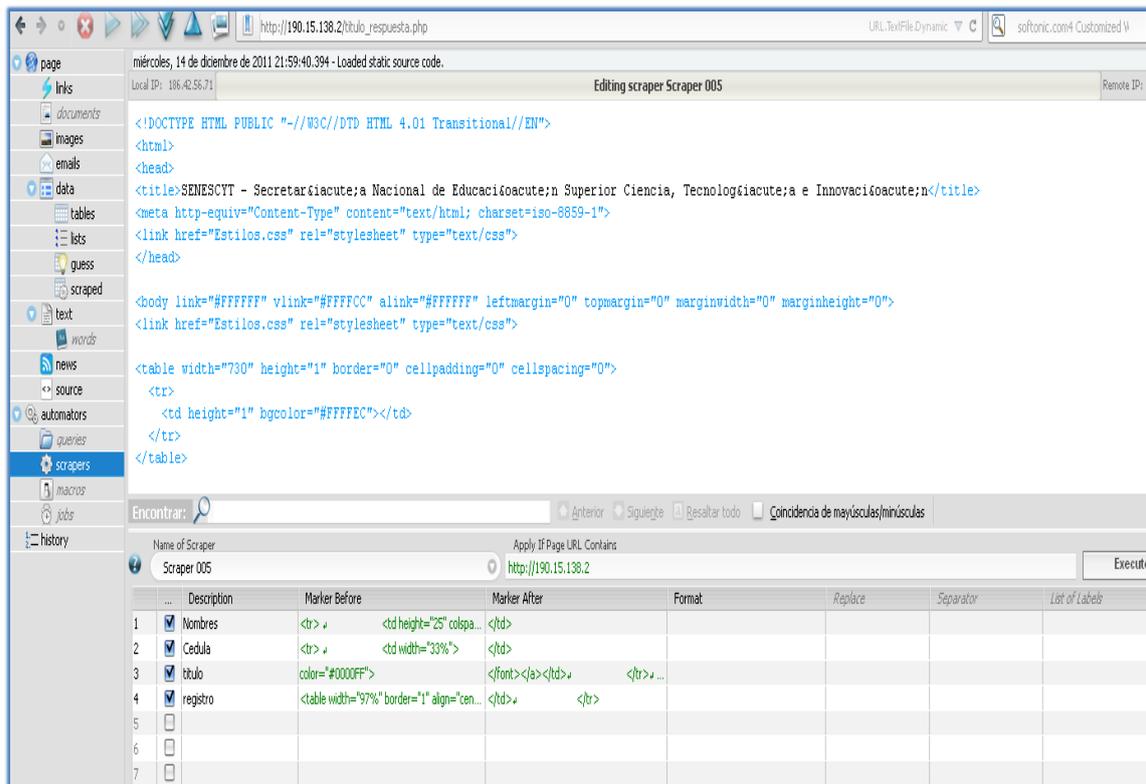


Figura19. Ventana principal del OUTWIT.

- 8.0- Luego de setear el scrapper se debe poner un nombre al mismo y guardarlo.
- 9.0- Se regresa a la opción links dando clic derecho opción Select all, luego clic derecho opción Auto-explore Selected links -> fast Scrape -> nuestro Scrapper (siendo este último el nombre del scrapper), comenzará a formarse el dataset de salida con las configuraciones en el Scrape.
- 10.0- Por último exportar el resultado al tipo de archivo deseado: SQL, CSV son algunos tipos de exportaciones posibles.

## Recopilación de Registros del SENESCYT.

Para la recopilación de los datos del antiguo Conesup hoy en día Senescyt, se recurre a la página donde se encuentran publicados y de dominio público a través de la siguiente dirección de internet que se encuentra, que es de libre acceso a consultas y de la cual se extraerá los datos, URL: [http://190.15.138.2/titulos\\_academicos.PHP](http://190.15.138.2/titulos_academicos.PHP) aquí se puede encontrar los títulos buscándolos por número de cédula o por apellidos, ya que los datos solo se muestran de forma unitaria en cuanto a las consultas a realizar, se extraerá los datos por medio de una herramienta de extracción de datos que es una extensión de FIREFOX, para minería de Datos de páginas web llamada "OUTWIT hub" en su versión light gratuita.

OUTWIT es una poderosa herramienta de recolección de datos, imágenes y muchos más datos que se pueden extraer de páginas web. Basada en búsquedas semánticas y una tecnología de reconocimiento que es considerada como "el primer paso hacia un buscador

semántico”.Justificamos el uso de OUTWIT pues al contar con esta herramienta basada en semántica nos aseguramos el utilizar al 100% estas tecnologías que se apegan sobre todo a los principios semánticos que pretendemos para nuestro proyecto.

Estos pasos para extraer los datos fueron realizados para todos los apellidos encontrados en la guía telefónica de Quito edición 2011, utilización que se justifica por el hecho del gran espectro de apellidos que podemos encontrar en la guía de teléfonos. En el caso de este proyecto los archivos fueron guardados como tipo CSV (delimitado por comas), y una vez completado el proceso para todos los apellidos encontrados se procede a unirlos en un solo archivo por medio del programa gratuito Beez File Merger en el cual escogemos todos los archivos a unir y la ruta de salida como el nombre del archivo de unión. Ahora que tenemos la muestra de registros volvemos a nuestra aplicación DRUPAL.

## Anexo B

### Instalación del CMS DRUPAL

#### Arquitectura de la aplicación

La arquitectura de la aplicación es una arquitectura web del tipo cliente-servidor en la que tenemos por un lado la aplicación en DRUPAL del lado del servidor y por el otro lado los posibles clientes que accedan a la misma para consumir los datos que se están como RDFS que se encuentran en él.

No se ha pretendido implementar una arquitectura con más de dos capas ni tampoco con una con web service en medio para tramitar las peticiones de los clientes, mas sin embargo nuestra arquitectura se justifica por el hecho de que las de este tipo son idóneas para implementaciones como esta en que las peticiones no necesitan estar en tiempo real. Por eso es suficiente una arquitectura clásica como la de cliente-servidor con la que trabaja DRUPAL. Se escoge DRUPAL como CMS para trabajar con este proyecto por ser el CMS de más acometida con respecto a la semántica y el Linked data que además es un Opensource que realmente busca el conocimiento para todos, no pretendiendo ser la mejor o competir con tecnologías Opensource de amplio espectro como SESAME y JENA en JAVA, nuevas tecnologías aplicadas de IBM y de Sun y otras de organismos gubernamentales, que podrían cubrir un abanico más grande de posibilidades en lo que ha semántica se refiere. El uso de DRUPAL se justifica en el hecho de que es un CMS 100% web PHP, JAVASCRIPT, HTML, css enmarcan claramente el concepto de Web.

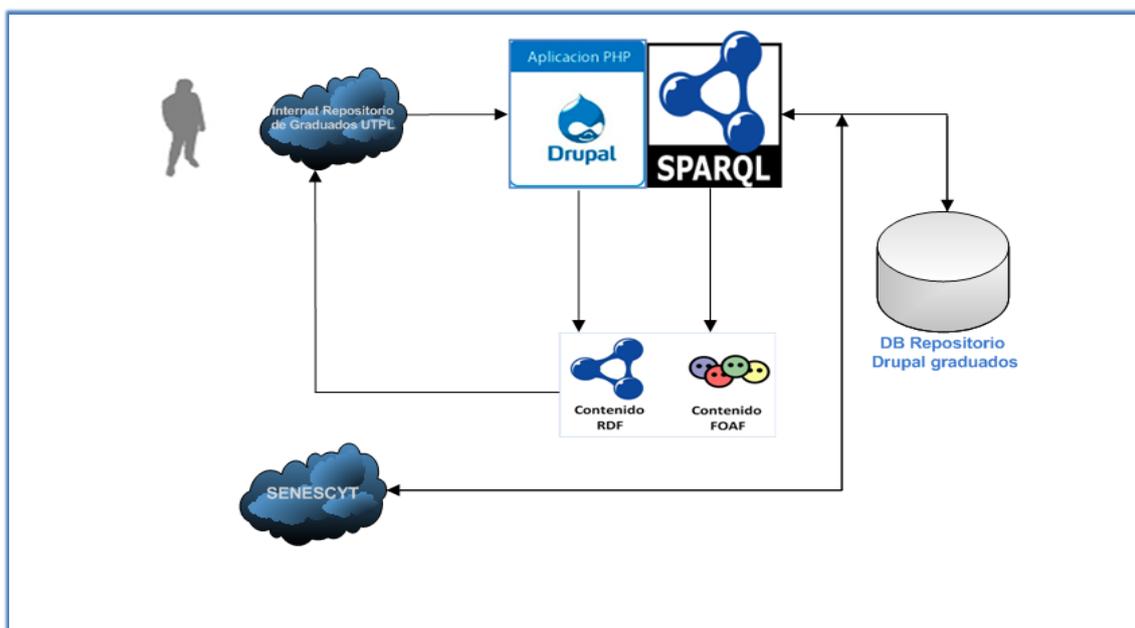


Figura20. Arquitectura de la implementación.

## Componentes

- DRUPAL 7.9
- Base de datos MYSQL
- Módulo RDFX
- Vocabularios: FOAF, RDFX, RDF, SIOC, SKOS, DC.
- Módulos para Vistas: VIEWS, SPARQLVIEWS.
- Módulo chart para gráficos.
- Librería RAPHAEL para gráficos.
- Módulo SPARQL para consultas al repositorio RDF y el SPARQL Endpoint.
- Otros módulos necesarios: ctools, entity, libraries.

## Requerimientos

Para instalar DRUPAL correctamente sobre una plataforma Windows, se debe cumplir ciertos requisitos antes de la instalación, tanto si se hace en localhost como en un hosting web:

- 1.- Tener montado un servidor apache, IIS o cualquier otro que atienda peticiones PHP. Para el caso del presente proyecto se utilizó un servidor apache inherente en el paquete xampp de libre descarga.
- 2.- Para la base de datos DRUPAL acepta MYSQL 5 o una superior, POSTGRES 8.3 y SQLLITE 3.x.
- 3.- PHP 5 o superior, que para el presente proyecto también fue obtenido del paquete xampp.
- 4.- Memoria RAM mínima 32M
- 5.- Un espacio en disco mínimo de 3 a 40 megas de acuerdo a la cantidad de módulos que se instalarán.
- 6.- Por último se necesita descargar el paquete de instalación de DRUPAL, desde supágina web.

Para la instalación desde un hosting el único requerimiento indispensable es el del punto 6 pues el resto de requerimientos los tienen por defecto la mayoría de hosting.

## Instalación

1.- Primero crear una base de datos para nuestro sitio al igual que su usuario y contraseña que tenga privilegios de administrador. Para crear nuestra base de datos ingresamos a nuestro PHPMYADMIN desde un browser, digitando la ruta estándar <http://localhost/PHPMYADMIN>.

1.1- Se busca el campo nueva base de datos y se introduce un nombre a la misma ej. "SENESCYT".

1.2- Se agrega un nuevo usuario a la base creada desde la parte de privilegios y se le otorga permisos de administrador.

El siguiente grafico muestra el panel de administración de PHPMYADMIN en la opción de crear nueva base de datos.

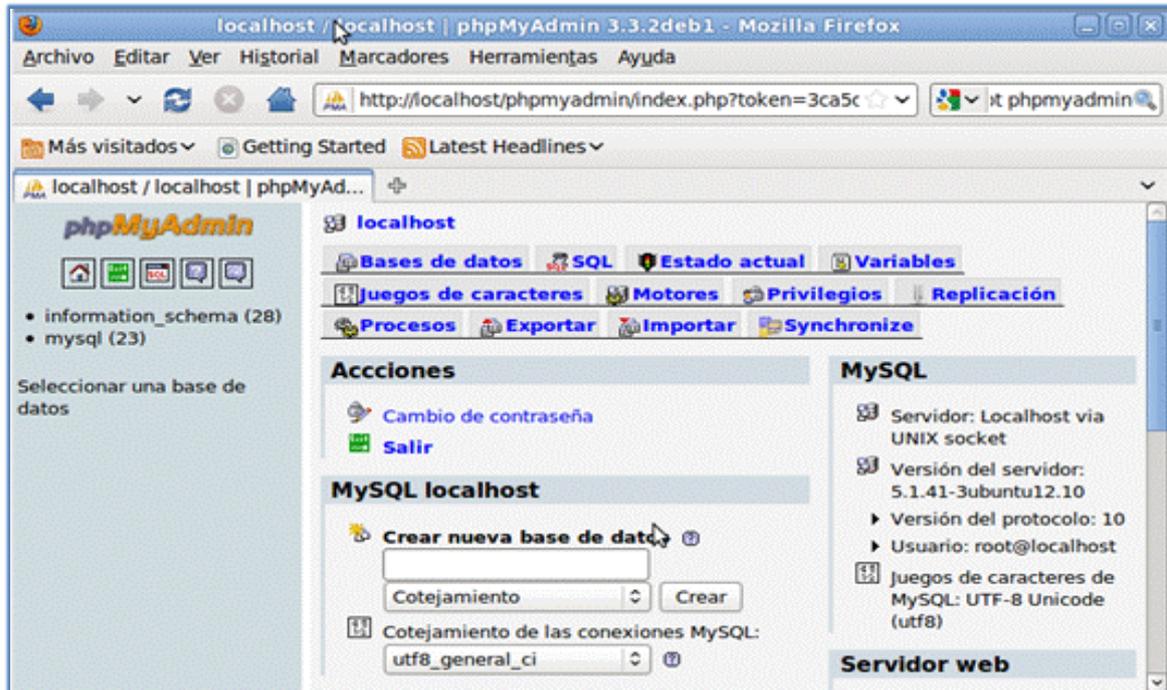
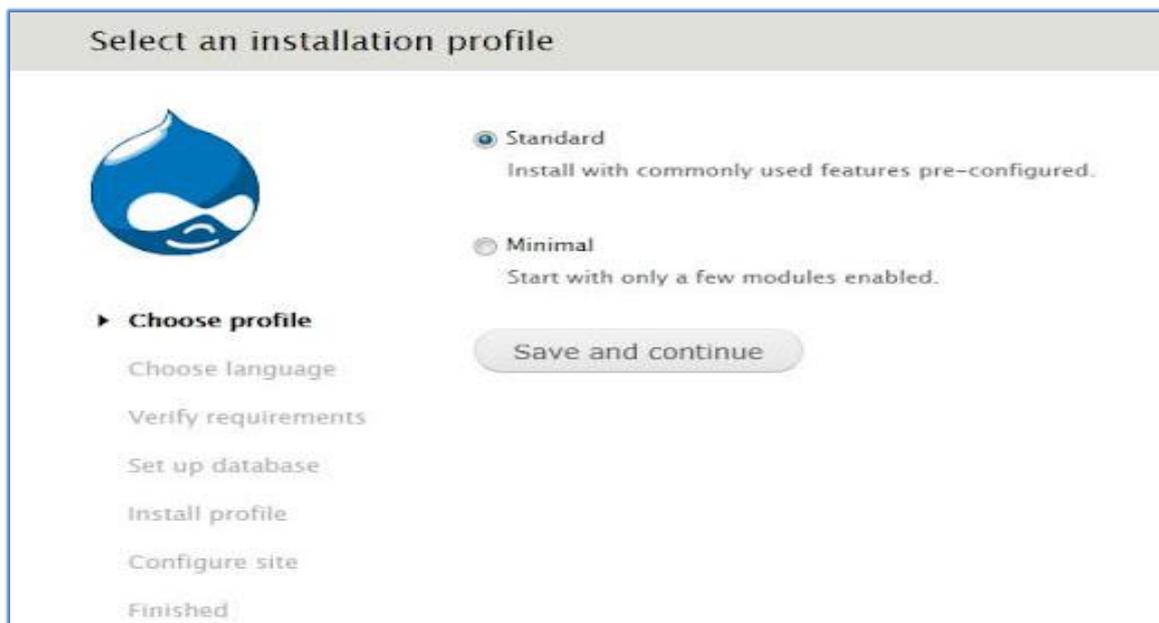


Figura21. Panel de PHPMYADMIN opción de crear base de datos.

2.- Se ingresaal localhost que automáticamente abrirá el PHP de instalación de DRUPAL para comenzar a setear el sitio, la dirección estándar del localhost es: http://localhost.com.

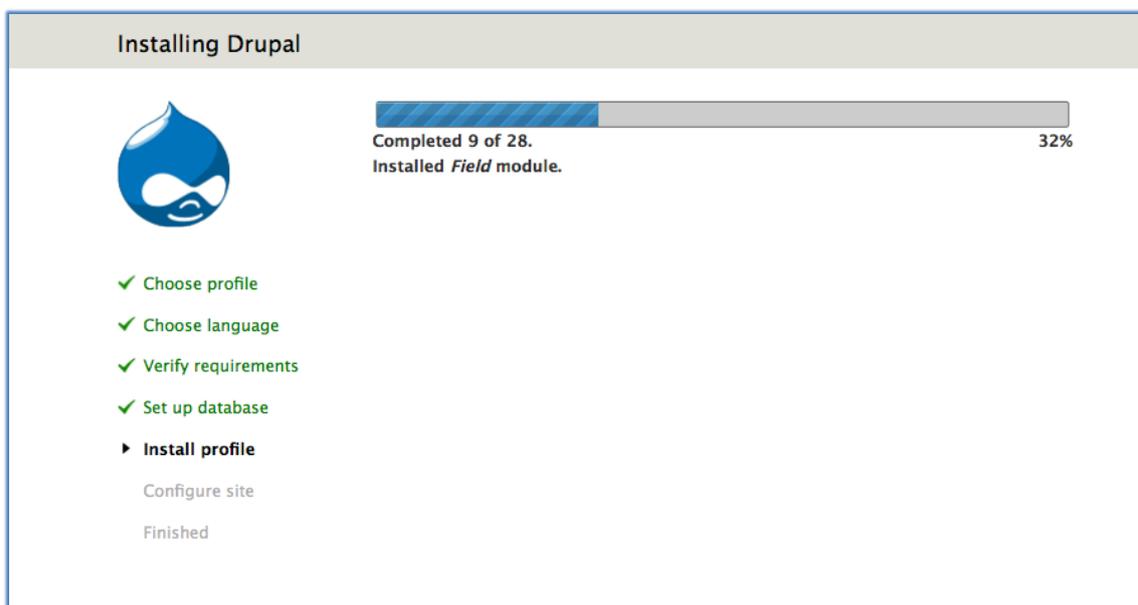


**Figura22. Opción de tipo de instalación.**

2.1 En la opción de tipo de instalación podremos escoger las características de DRUPAL que deseamos instalar en nuestro sitio, se recomienda instalar la versión estándar y luego editar en la opción de modules si deseáramos agregar o quitar algún componente, Save and Continue.

2.2 En la siguiente página se elige el idioma deseado, español para este ejemplo > Save and Continue.

2.3- La página siguiente es la de configuración de base de datos, aquí escogeremos el tipo de base de datos que utilizara DRUPAL que como vimos anteriormente es MYSQL, ingresamos el nombre de la base de datos que creamos "SENESCYT", el usuario y contraseña > Save and Continue.



**Figura23. Ventana de configuración de base de datos.**

3.- Una vez instalados los módulos de DRUPAL aparecerá la página de configuración del sitio, ingresamos los datos del administrador como es el nombre de usuario y contraseña, el mail, seleccionamos el país en donde se encuentra este sitio y su zona horaria y guardamos estas configuraciones.

Una vez todas estas configuraciones estén correctamente guardadas y validadas la siguiente pantalla que aparecerá es la del nuevo sitio creado, con un menú en la parte superior donde se verá este nombre de usuario y los diferentes tipos de menús que posee DRUPAL para las configuraciones.

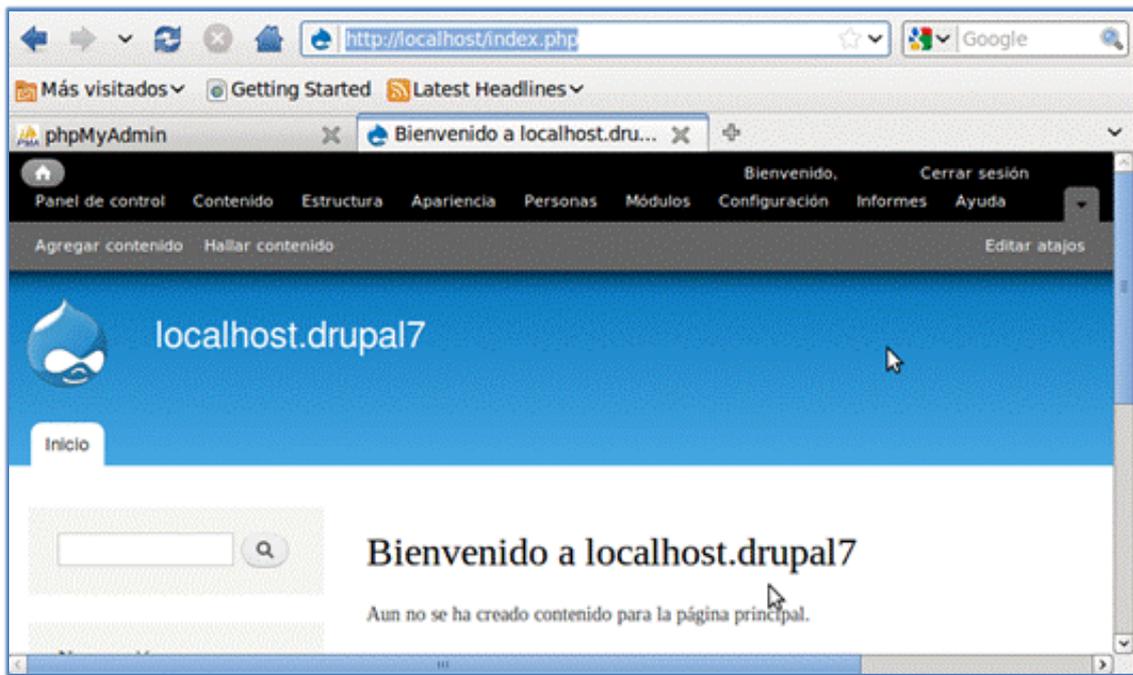


Figura24. Grafico del home del sitio

## Habilitación de módulos y corrección de errores

La habilitación de módulos se hace desde la consola de administración escogiendo la opción MODULES, otra manera de hacerlo es ingresando la siguiente Url. <http://syberiaec.com/#overlay=admin/modules> allí se encuentran los módulos que por defecto trae DRUPAL, así como los módulos que uno instala después. Escogiendo la opción “install new module” usted puede instalar un nuevo módulo que se encuentre en su pc o a través de una Url que usted conozca que contiene uno.

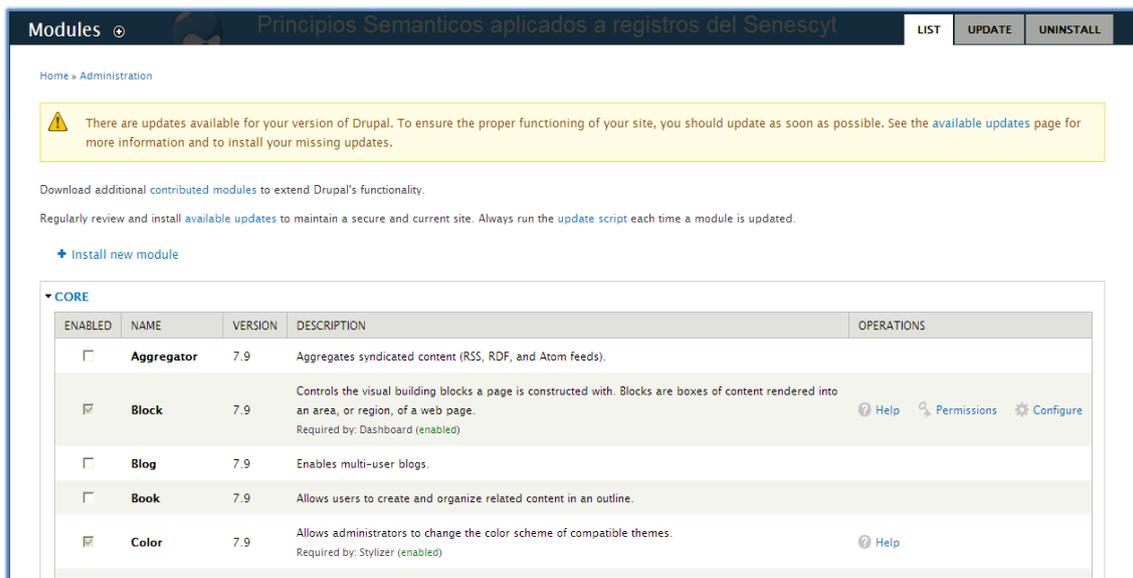
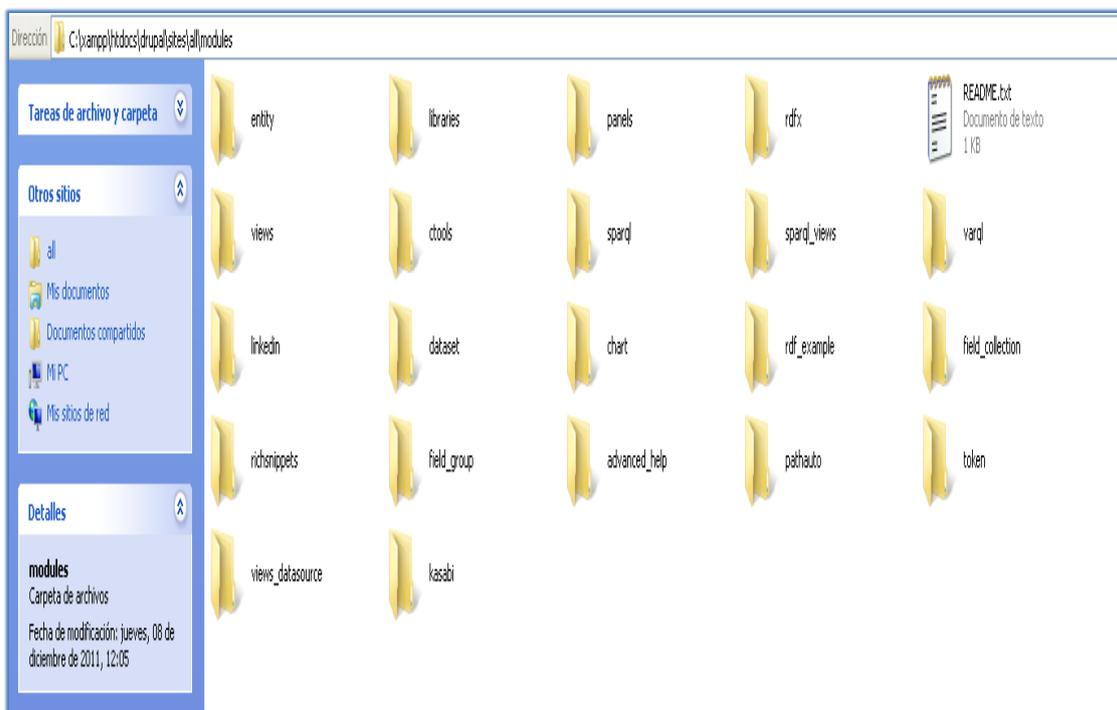


Figura25. Administración de módulos en DRUPAL.

En ocasiones algunos módulos no se instalan directamente desde la administración del sitio entonces otro método para instalarlos es colocándolos dentro de la carpeta sites/all/ modules que se encuentra dentro de la carpeta de instalación de DRUPAL.



**Figura26. Instalación de módulos a través de directorio.**

Luego de instalarlos o copiarlos se deben habilitar desde el panel de administración → opción modules seleccionando el casillero de los mismos para que se active y procedemos a guardarlos mediante un clic en la opción “save”. Una vez habilitados los módulos revisaremos ingresando a la opción reportes para ver si existe algún error reconocido por DRUPAL con respecto a la instalación de los módulos de ser así se deberá buscar en la documentación y foros de DRUPAL el fix correspondiente, en esta implementación se presentaron 6 bugs a los que se aplicaron parches de colaboradores de la comunidad DRUPAL, los bugs son muy específicos y dependen de muchos factores como el servidor en el que se está trabajando los seteos iniciales las restricciones del hosting y otros como la versión del CMS, por lo que no se los ha documentado por ser demasiado particulares y no se los podría abarcar en su totalidad.

Para la aplicación deberemos instalar y habilitar los módulos siguientes: SPARQL, ctools, RDFX, entity, VIEWS, SPARQL\_VIEWS, library, chart.

## Importar Vocabularios

En la página principal de este sitio tendremos algunas opciones en el bloque de NAVIGATION entre ellas la de “IMPORT CORE RDF VOCABULARIES”, escogemos esta opción y le damos un clic. Si todos los bugs de los módulos que instalamos están parchados se importaran los vocabularios más conocidos y todos sus propiedades y clases estarán listas para nuestras

configuraciones RDF. Los vocabularios que se importan son: DC, FOAF, EDM, RDFS, SIOC, SIOCT, SKOS, XSD. Existen más vocabularios que no se importan pero que también podemos trabajar con ellos importándolos individualmente mediante la opción "IMPORT EXTERNAL VOCABULARY".

## **Inserción de los datos a la base del repositorio**

Se crea un nuevo tipo de contenido en este sitio y se procede a crear campos como: nombres, cédula, título, registro y nacionalidad, que luego serán configurados para que los mismos sean utilizados como datos RDF. Se insertan los registros recopilados, para este proyecto 44.000 registros extraídos de la SENESCYT directamente a la base de datos para que puedan verse posteriormente como un nodo de DRUPAL.

Tablas involucradas en la Inserción:

- Node
- Node\_revision
- node\_comment\_statistics
- Field\_data\_field\_cédula (cédula)
- Field\_revision\_field\_cédula (cédula)
- Field\_data\_field\_nombres (nombres)
- Field\_revision\_field\_nombres (nombres)
- Field\_data\_field\_título (título)
- Field\_revision\_field\_título (título)
- Field\_data\_field\_nacionalidad (nacionalidad)
- Field\_revision\_field\_nacionalidad (nacionalidad)
- Field\_data\_field\_registro (registro)
- Field\_revision\_field\_registro (registro)

## Tablas del repositorio

	nid	vid	type	language	title	uid	status	created	changed	comment	promote	sticky
	The primary identifier for a node.	The current node revision version identifier.	The type of this node.	The language of this node.	The title of this node, always treated as non-markup text.	The user id that owns this node; initially, this is the user that created it.	Boolean indicating whether the node is published (visible to non-administrators).	Unix timestamp when the node was created.	Unix timestamp when the node was most recently saved.	Whether comments are allowed on this node: 0 = no, 1 = closed (read only), 2 = open (read/write).	Boolean indicating whether the node should be displayed on the front page.	Boolean indicating whether the node should be displayed at the top of lists in which it appears.
Editar	1	1	titulos	und	titulo	1	1	1323575147	1323575147	2	1	0
Editar	2	2	titulos	und	titulo	1	1	1323578269	1323578269	1	1	0
Editar	3	3	titulos	und	titulo	1	1	1323578329	1323578329	1	1	0
Editar	7	7	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	8	8	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	9	9	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	10	10	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	11	11	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	12	12	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	13	13	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	14	14	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	15	15	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	16	16	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	17	17	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	18	18	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	19	19	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	20	20	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0
Editar	21	21	titulos	und	titulo	1	1	1323578375	1323578375	1	1	0

Figura27. Estructura de las tablas del Repositorio.

1.0- Se procede a la importación de los datos extraídos hacia cada una de las tablas de la base de datos para la inserción a través del administrador de base de datos MYSQL “PHPMYADMIN” de este sitio manteniendo la estructura de cada tabla en cuestión.

Importando en la tabla "node"

Archivo a importar:

El archivo puede ser comprimido (gzip, zip) o descomprimido.  
Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador:  Examinar... (Máximo: 256MB)

Conjunto de caracteres del archivo: utf-8

Importación parcial:

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.)

Número de filas a omitir, iniciando de la primera fila:

Formato:

SQL

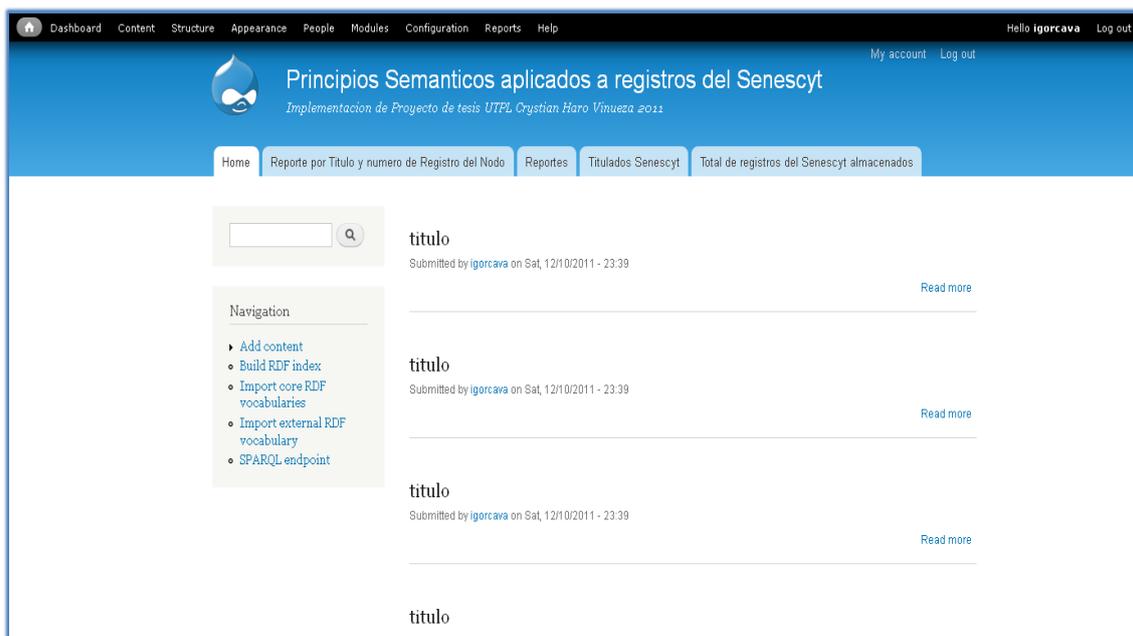
Opciones específicas al formato:

Modalidad SQL compatible: NONE

[http://www.syberiaec.com:2082/3rdparty/phpMyAdmin/tbl\\_import.p...beriae\\_conep&table=node&token=cf046b7cd59e43096ce7133](http://www.syberiaec.com:2082/3rdparty/phpMyAdmin/tbl_import.p...beriae_conep&table=node&token=cf046b7cd59e43096ce7133)

Figura28. Módulo de importación del PHPMYADMIN.

Terminado este proceso de inserción en la base de datos ya podemos ver los nodos en este sitio listos para trabajar con ellos.



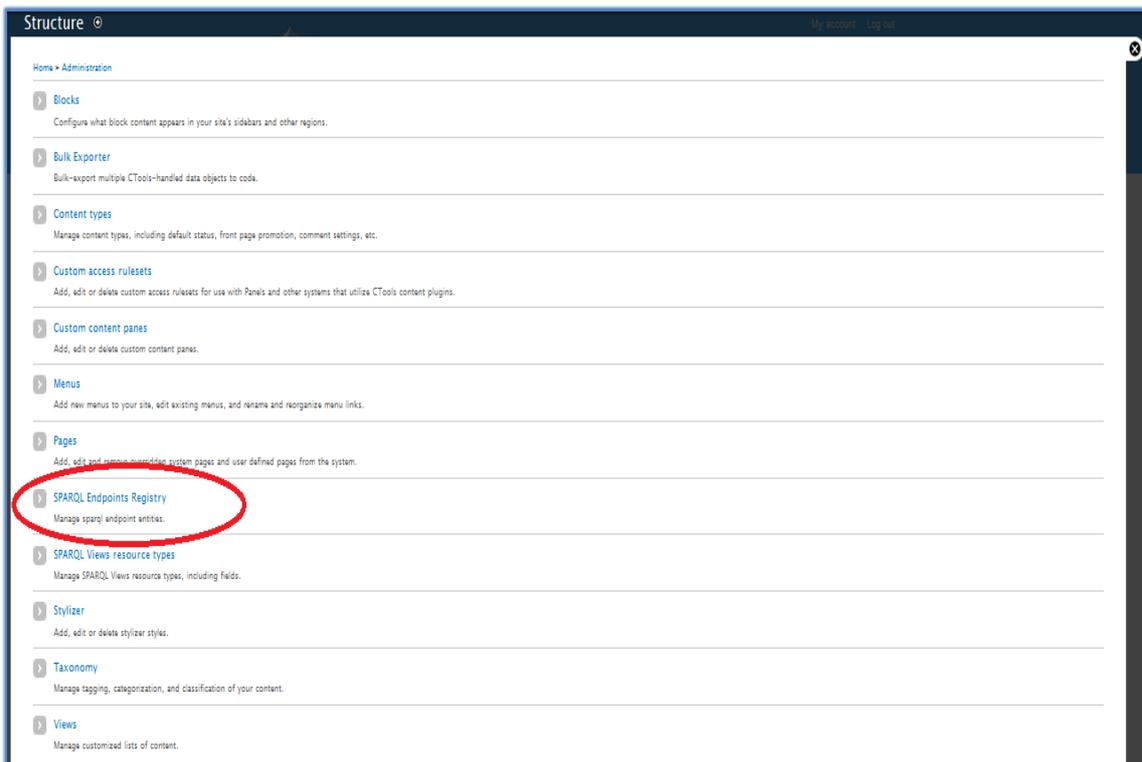
**Figura29. Nodos agregados resultado de la reingeniería.**

Para que todos los registros ingresados en la base estén dentro del RDF es necesario indexarlos para que los mismos se encuentren dentro del store y puedan ser seleccionados a través de los QUERYSPARQL y del SPARQL endpoint, para reconstruir estos índices tenemos que escoger la opción “build RDF índice” que se encuentra en la sección de navigation de la página principal y automáticamente todos los índices RDF serán reconstruidos.

## Inserción de los datos a la base del repositorio

Para la creación de vistas que ocupará el módulo VIEWS y SPARQL necesitaremos descargar una librería necesaria que se llama ARC2, dicha librería se la puede encontrar para descarga libre. Una vez descargada la librería la colocamos su contenido en el directorio sites/all/module/libraries /vendor/arc/si no existe esta ruta crearemos manualmente los directorios vendor/arc. Luego de tener la librería ARC lista, se construyen las diferentes visualizaciones de los Nodos internos y contenido externo por medio del SPARQL\_VIEWS, módulo que permite crear vistas a partir de RDF endpoints y visualizar la data contenida en forma de grillas gráficos o formato HTML, la utilización de este módulo se justifica por su diversidad de salidas interacción con endpoints internos y externos, además de su versatilidad y compatibilidad con otros módulos como el chart para gráficos.

Para crear la vista se debe ingresar a structure→VIEWS→add new view para crear una vista de contenido tipo RDF “Título” así como una vista tipo SPARQL\_VIEWS que permitirá trabajar con dataset propios asignados en endpoint internos y externos, en este sitio el SPARQL-endpoint es: <http://www.syberiaec.com/sparql> y también escoger si queremos que la vista se muestre en una grilla u otro formato de salida como el de una gráfica. Primero se ingresa a Structure y escoge la opción SPARQL endpoints registry.



**Figura30. Opción SPARQLEndpoints Registry.**

Luego se añade un Endpoint que puede ser un dataset o simplemente una URI y se le da un nombre, este Endpoint es la URI donde se encuentran los recursos RDF disponibles para que se pueda extraer la información, en este caso se le da la URI de Endpoint local pero podría ser un Endpoint externo.



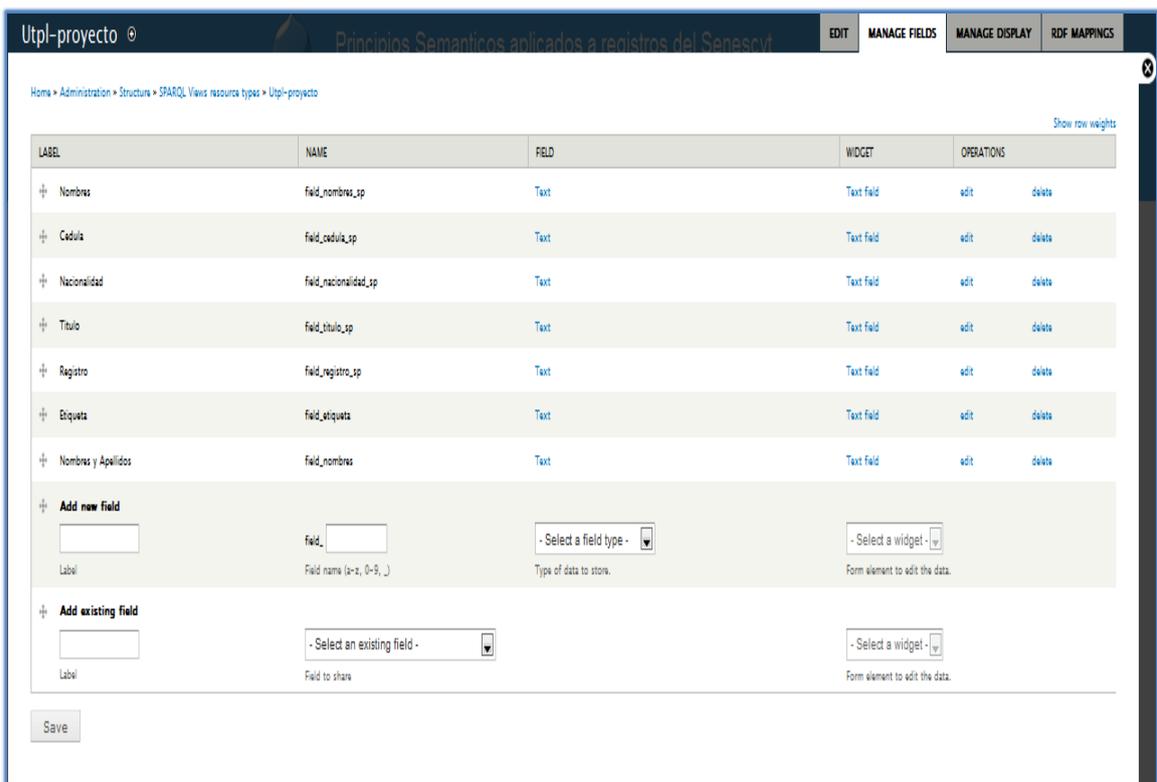
**Figura31. SPARQL Endpoints añadidos al sitio.**

Paso seguido se configuran los tipos de recursos que se pretende sacar de este Endpoint a través de las vistas SPARQL. Este paso se lo realiza ingresando a Structure→SPARQLVIEWS resource types.



**Figura32. Configuración de tipos de contenidos del endpoint.**

Se le asigna un nombre y se añade campos con nombres referentes a los datos RDF que se desea extraer.



**Figura33. Campos añadidos al registro del endpoint.**

Y se le asigna las referencias RDF con las que están creadas. Por ejemplo para el campo nombre el mapeo RDF será FOAF:NAME pues ese es el tipo de recurso que representa la etiqueta Nombres.

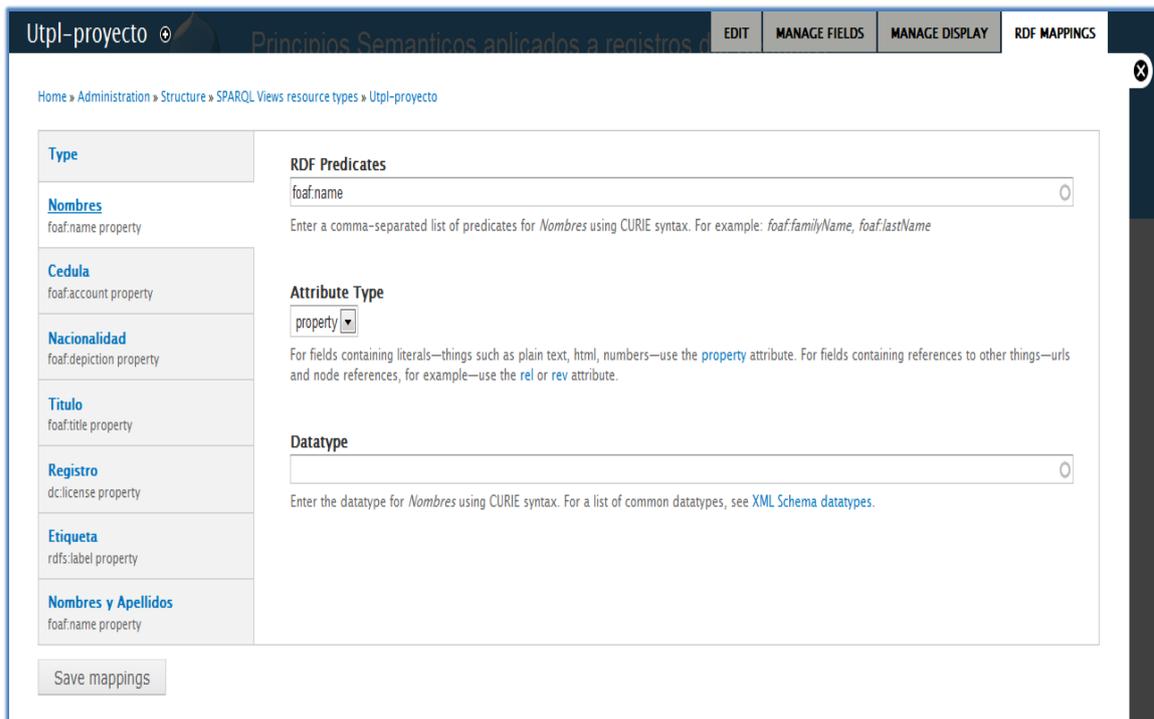


Figura34. Mapeos RDF para cada tipo de contenido.

## Verificación del SPARQL Endpoint

El módulo SPARQL endpoint cuenta con una interfaz para consultas QUERYS por parte del usuario con respecto a las URIS y el contenido RDF, que nos permite hacer consultas internas y externas del contenido así como también elegir el formato de salida preferido, como por ejemplo: RDF/XML, JSON etc.

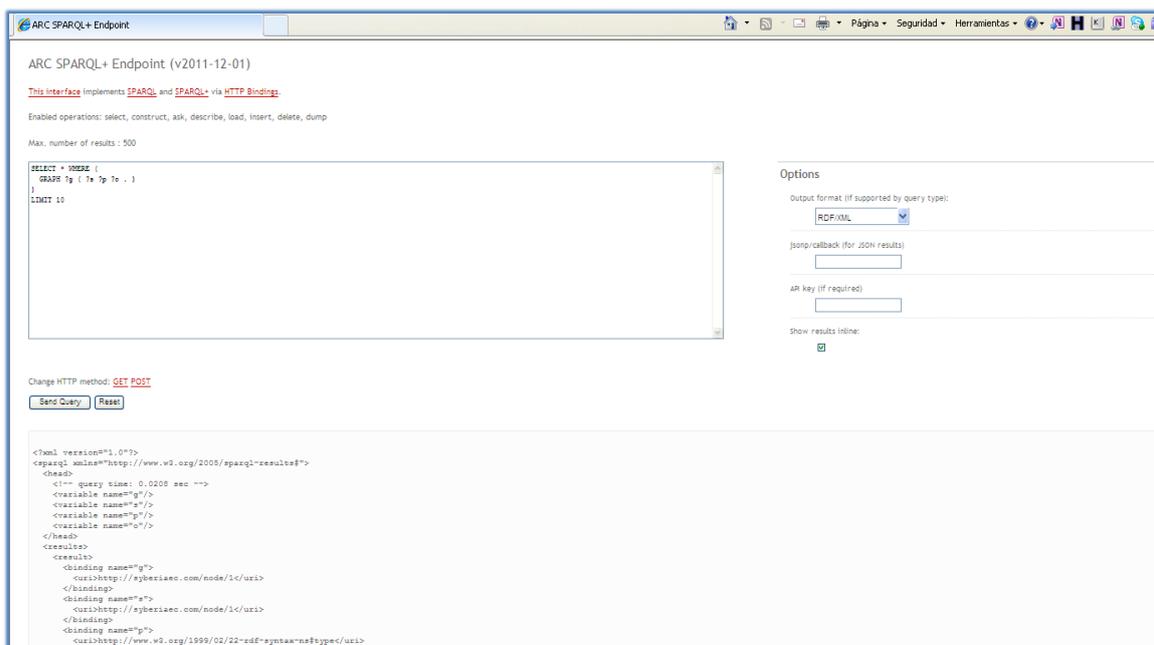


Figura35. Endpoint de SPARQL.

Se puede sacar los datos por medio de consultas SPARQL desde el SPARQL Endpoint. Y hacer select de los Datos Almacenados en RDF, aquí un ejemplo de consulta que filtra el nombre "Cristian Antonio Haro Vinuesa" que busca si consta como un valor dentro de la propiedad FOAF:NAME.

# Glosario

---

## **ALLEGROGRAPH**

Es una moderna y de alto rendimiento base de datos persistente grafo RDF. ALLEGROGRAPH utiliza el almacenamiento basado en disco, lo que le permite escalar a miles de millones de triples, mientras que mantiene un rendimiento superior. ALLEGROGRAPH apoya SPARQL, RDFS + +, y el razonamiento prólogo de numerosas aplicaciones cliente.

## **ASK**

Función que retorna un valor booleano indicando si los patrones de consultas fueron coincidentes.

## **Bnodes**

BNodes sirve al propósito de representar los recursos que no pueden ser denominados en la actualidad. Por ejemplo, bNodes permiten representar algo como "alguien cuyo nombre es 'Juan' y que tiene 25 años".

## **CMSDRUPAL**

DRUPAL es un contenido de código abierto plataforma de gestión de alimentación de millones de sitios web y aplicaciones. Está construido mayormente en lenguaje PHP y cuenta con el apoyo de una comunidad activa y diversa de personas en todo el mundo.

## **CodeMetrics**

Las métricas de código es un conjunto de medidas de software que proporcionan a los desarrolladores una mejor visión del código que están desarrollando. Al tomar ventaja de métricas de código, los desarrolladores pueden entender qué tipos y / o métodos se deben rehacer o probar más a fondo. Los equipos de desarrollo pueden identificar riesgos potenciales, comprender el estado actual de un proyecto, y seguir el progreso durante el desarrollo de software.

## **Conesup**

El Consejo Nacional de Educación Superior.Sus servicios son la legalización de firmas de documentos universitarios, que es un requisito solicitado en instituciones internacionales, lo cual es necesario para personas que viajan a exterior. Certificaciones de títulos, necesario para ejercer la profesión, ingresar al Colegio profesional y legalizaciones para viajes fuera del país, esto se aplica para títulos de técnico superior, tercero y cuarto nivel sea de profesionales ecuatorianos o extranjeros. Además emitirá un informe para proyectos de universidades, alaAsamblea Nacional quien expedirá la ley de creación de dicha universidad.

## **CONSTRUCT**

Función Que devuelve un grafo RDF construido por laSustitución de variables en un conjunto de tres plantillas, para luego combinarlo en una sola tripleta RDF.

## **DAV**

Librería de JAVASCRIPT que permite crear y generar poderosos gráficos.

## **DAV**

Provee un WebDAV.

## **DBMS**

Los sistemas de gestión de bases de datos (en inglés data base management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

## **DRUPAL**

DRUPAL es un CMS de código abierto con licencia GNU/GLP especialmente diseñado para construir comunidades en internet.

## **Dublín Core**

Dublín Core es un modelo de metadatos elaborado y auspiciado por la DCMI (Dublín Core Metadata Initiative), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos.

## **Dublín de JAVA**

El de Dublín JAVA Meetup Group (DJMG) es una iniciativa para conectar a la gente de Irlanda interesados en las tecnologías JAVA. Se trata de un evento mensual organizado por Jakob Korab. La descripción de los objetivos de la reunión se puede encontrar en la página web DJMG.

## **FeedAPIRDF Processor**

Guarda ítems del feed en formato RDF.

## **FOAF**

Documento de vocabulario técnico, describe y especifica cada elemento del espacio de nombres.

## **Framework**

Un conjunto de bibliotecas reutilizables o clases de un sistema de software (o subsistema).

## **Google**

Buscador que enfoca sus resultados para este país y a nivel internacional tanto en español como en inglés.

## **Google Chart API**

Herramienta de google que sirve para la creación de Gráficas.

## **JENA**

JENA es un framework JAVA para la creación de la Web Semántica aplicaciones. Proporciona un entorno de programación para RDF, RDFS y de OWL, SPARQL e incluye una base del motor de inferencia de la regla.

## **Opensource**

Es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

## **OWL**

Es el acrónimo del inglés Ontology Web Language, un lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML.

## **PREFUSE**

Toolkit en JAVA para creación de visualizaciones.

## **RAPHAEL**

Librería de JAVASCRIPT que ayuda a la elaboración de Gráficas.

## **RDF**

Sigla del inglés "Resource Description Framework", Marco de Descripción de Recursos, un lenguaje de descripción del W3C.

## **Relations API**

Provee un API para una relación arbitraria de nodos basados en RDF.

## **RSS**

Son las siglas de Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador).

## **SENECYT**

Secretaria Nacional de Educación Superior, Ciencia, Tecnología e Innovación

## **SELECT**

La sentencia SELECT nos permite consultar los datos almacenados en una tabla de la base de datos.

## **SERQL**

SERQL (SESAMERDFQUERY Language, pronunciado como "circle") es un lenguaje de recuperación para RDF/RDFS desarrollado por Aduna como parte del software SESAME. Combina características de otros lenguajes (principalmente RQL, RDQL, N-Triples y N3) y añade otras propias.

## **SESAME**

Es un framework de la código Abierto para preguntar y analizar datos de RDF. Fue creado, y todavía está siendo mantenido por la empresa de informática holandesa Aduna. Fue desarrollado originalmente como parte del "En-A-Conocimiento", un proyecto semántico de la tela que funcionó a partir la 1999 a 2002. Contiene un triplestore.

## **SIOC**

Comunidades online interconectadas semánticamente.

## **SKOS**

Lenguaje RDF que brinda un modelo para representar estructuras y contenidos de esquemas.

## **SPARQL**

(SPARQL Protocol And RDFQUERY Language, "sparkle" en español) definen un lenguaje de recuperación de información para RDF/RDFS. Con SPARQL no sirve para no preocuparnos por la tecnología utilizada en la base de datos. Pues las consultas en el lenguaje SPARQL expresan objetivos de alto nivel, es fácil extenderlos a orígenes de datos inesperados, o incluso transferirlos a nuevas aplicaciones.

## **SPARQL Endpoint**

Recurso tipo URI que contiene información semántica del tipo dataset y que permite hacer consultas SPARQL.

### **Topic Map**

Los Topic Maps son un estándar que se está implantando en el web y que posibilita la navegación conceptual.

### **TURTLE**

Es un formato de serialización de recursos RDF.

### **URIS**

Un URI (Uniform Resource Identifier) se diferencia de un URL en que permite incluir en la dirección una subdirección, determinada por el "fragmento". Esto se comprende mejor analizando la estructura de un URI.

### **VIEWS DataSource**

Es un conjunto de pluggins para Vistas de DRUPAL para dar nodos de contenido en un número de compatible y formato reusable basado en XML, JSON y XHTML.

### **W3C**

El W3C desarrolla estándares, técnicas y directrices enfocadas a mejorar la calidad de las aplicaciones web.

### **Wikipedia**

Autodefinida como un esfuerzo colaborativo por crear una enciclopedia gratis, libre y accesible por todos. Permite revisar, escribir y solicitar artículos.

### **WorldWideWeb**

La World Wide Web es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

### **XML**

Es un lenguaje extensible de etiquetas desarrollado por el W3C.

# Índice Alfabético

---

## **A**

ARC2 · 48

---

## **Ch**

chart · 41, 45, 48

---

## **C**

ctools · 41, 45

---

## **D**

dataset · 37, 38, 48, 49, 57

---

## **E**

entity · 41, 45

---

## **F**

FIREFOX · 38

FOAF · 54, 60

---

## **H**

HTML · 11, 40, 48

---

## **J**

**JAVASCRIPT** · 40

JSON · 51, 57

---

## **L**

lenguaje · 4, 55, 56, 57

Linked data · 34, 40

---

## **M**

MODULES · 44

---

## **O**

OUTWIT · 38

OWL · 4, 11, 55

---

## **P**

**PHP** · 40

PHPMYADMIN · 47

---

## **R**

RDF · 4, 12, 18, 34, 45, 46, 48, 49, 50, 51,  
53, 54, 55, 56

RDFX · 41, 45

Registro · 23

repositorio RDF · 6, 41

---

## **S**

**Select** · 38

**Senescyt** · I, II, XI, 18, 20, 25, 26, 38

SIOC · 10, 12, 41

SKOS · 10, 11, 12, 41

SPARQL · 12, 53, 55, 56, 57, 60

SPARQL Endpoint · 41, 52

---

## **T**

tabla · 47, 56

---

## **U**

URI · 57

---

**V**

VIEWS, SPARQL · 41, 45

Vista · 23, 24, 25, 26, 27, 28, 30

vocabulario · 11, 12, 13, 54

---

**W**

W3C · 57, 61

Wikipedia · 57, 60, 61

---

**X**

XML · 51, 55, 56, 57

# Bibliografía

---

[1] “La Web Semántica: fundamentos y breve "estado del arte"” Luis Sánchez Fernández, (2009) Disponible en:

<http://www.ati.es/novatica/2005/178/178-6.pdf>

[2] “Web Semántica” Wikipedia (2012) Disponible en:

[http://en.wikipedia.org/wiki/Semantic\\_Web](http://en.wikipedia.org/wiki/Semantic_Web)

[3] “La web semántica y el vocabulario RDF”, Garland Foster (2011) Disponible en:

<http://www.malditainternet.com/2001/02/21/la-web-semantica-y-el-vocabulario-RDF/>

[4] “Lenguaje de Ontología web OWL” W3C (2012) Disponible en:

<http://www.w3.org/TR/OWL-features/>

[5] “Esquemas RDF”, María Lamarca (2010) Disponible en:

[http://www.hipertexto.info/documentos/esquemas\\_RDF.htm](http://www.hipertexto.info/documentos/esquemas_RDF.htm)

[6] “Esquemas RDF”, María Lamarca (2010) Disponible en:

[http://www.hipertexto.info/documentos/esquemas\\_RDF.htm](http://www.hipertexto.info/documentos/esquemas_RDF.htm)

[7] “Repositorios RDF “, W3C (2010) Disponible en:

[http://www.w3.org/egov/wiki/RDF\\_Repository](http://www.w3.org/egov/wiki/RDF_Repository)

[8] “EuropeanaRDFStore Report “, Bernhard Haslhofer, Elaheh Momeni, Bernhard Schandl, Stefan Zander (2011) Disponible en:

[http://www.europeanaconnect.eu/documents/europeana\\_ts\\_report.pdf](http://www.europeanaconnect.eu/documents/europeana_ts_report.pdf)

[9] “SPARQLQUERY Language for RDF”, W3C (2008) Disponible en:

<http://www.w3.org/TR/RDF-SPARQL-QUERY/>

[10] “DtDs y XML Esquema”, Lamarca María Jesús (2009) Disponible en:

<http://www.hipertexto.info/documentos/dtds.htm>

[11] “FOAF Vocabulary Specification”, W3C (2007) Disponible en:

<http://XMLns.com/FOAF/0.1/>

[12] “Dublín Core “, Wikipedia (2012) Disponible en:

[http://es.wikipedia.org/wiki/Dublin\\_Core](http://es.wikipedia.org/wiki/Dublin_Core)

[13] “Simple Knowledge Organization System”, Pastor José (2011) Disponible en:

[ggomez.files.wordpress.com/2008/09/informe-SPARQL.doc](http://ggomez.files.wordpress.com/2008/09/informe-SPARQL.doc)

- [14] “Manual de SKOS”, Pastor José (2009) Disponible en:  
<http://SKOS.um.es/TR/SKOS-primer/>
- [15] “SIOC”, W3C (2010) Disponible en:  
<http://semanticweb.org/wiki/SIOC>
- [16] “RDF y RDF schema”, García Grecia, (2003) Disponible en:  
[http://www.matem.unam.mx/~grecia/semantic\\_web/RDF.html](http://www.matem.unam.mx/~grecia/semantic_web/RDF.html)
- [17] “IHMC Cmap Tools”, IHMC (2010) Disponible en:  
<http://cmap.ihmc.us/>
- [18] “Visualización de Metadatos“, Herskovic Valeria (2010) Disponible en:  
[https://www.u-cursos.cl/ingenieria/2006/1/CC70W/1/material\\_docente/objeto/96043+&cd=1&hl=es&ct=clnk&q=ec](https://www.u-cursos.cl/ingenieria/2006/1/CC70W/1/material_docente/objeto/96043+&cd=1&hl=es&ct=clnk&q=ec)
- [19] “API de Google para creación de gráficas”, Álvarez Ángel, (2009) Disponible en:  
<http://www.desarrolloweb.com/articulos/API-google-para-creacion-de-gráficas.html>
- [20] “Google Chart Tools” Google (2011) Disponible en:  
<http://code.google.com/intl/es/APIS/chart/>
- [21] “API de Google para creación de gráficas”, Álvarez Ángel (2009) Disponible en:  
<http://www.desarrolloweb.com/articulos/API-google-para-creacion-de-gráficas.html>
- [22] “The PREFUSE visualization toolkit”, GITHUB (2012) Disponible en:  
<http://PREFUSE.org/>
- [23] “Data-Driven Documents”, GITHUB (2012) Disponible en:  
<http://mbostock.github.com/d3/>
- [24] “Raphaël—JAVASCRIPT Library”, Baranovskiy Dimitri (2011) Disponible en:  
<http://RAPHAELjs.com/>
- [25] “Conjuntos de Datos, Vocabularios controlados y Conjuntos de Elementos de Metadatos”, Antoine Isaac, Europeana and VrijeUniversiteitAmsterdam, Netherlands  
William Waites, University of Edinburgh (School of Informatics), UK  
Jeff Young, OCLC Online Computer Library Center, Inc., US  
Marcia Zeng, Kent State University, (2011) Disponible en:  
<http://SKOS.um.es/Incubator/IIId/XGR-IIId-vocabdataset/>
- [26] “Technical Tools and Standards of the Semantic Web”, Culley AI, (2006) Disponible en:  
<http://www.instructionaldesign.com.au/Academic/TechnicalTheme1.htm>

[27] “Anotaciones Semánticas”, Blanco S, (2007) Disponible en:

<http://www.infor.uva.es/~sblanco/Tesis/Anotaciones%20Semánticas.pdf>

[28] “RDF Vocabulary Description Language”, Brinckley Dan, Guha R.V (2004) Disponible en:

<http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>

[29] “De la web actual a la web semántica”, Marlon Alberto Piñeres Melo Mg, Ilma Azucena Bonilla Botía Mg (2008) Disponible en:

[http://www.uac.edu.co/images/stories/publicaciones/revistas\\_cientificas/prospectiva/volumen-6-no-2/articulo9-v6n2.pdf](http://www.uac.edu.co/images/stories/publicaciones/revistas_cientificas/prospectiva/volumen-6-no-2/articulo9-v6n2.pdf)

[30] “Web convencional VS Web semántica”, ecarhoh (2012) Disponible en:

<http://ecarhoh.wordpress.com/2012/02/08/web-convencional-vs-web-semantica/>

[31] “Diseño de un repositorio RDF basado en tecnologías NOSQL”, González, Illarramendi y Bermúdez (2012) Disponible en:

[http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S1/Emergentes/JISBD2011\\_articulo.pdf](http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S1/Emergentes/JISBD2011_articulo.pdf)