



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“ANÁLISIS DE LAS TECNOLOGIAS RDF, OWL, SWRL EN LA WEB**  
**SEMÁNTICA PARA CREAR ONTOLOGÍAS DE CURSOS**  
**VIRTUALES A PARTIR DE BASE DE DATOS DE UN LMS”**

Trabajo de titulación presentado para optar al grado académico de:  
**INGENIERA EN SISTEMAS INFORMÁTICOS**

**AUTORA:** DALILA LUCÍA TOLEDO SANMARTÍN  
**TUTOR:** Ms. DANILO MAURICIO PASTOR RAMIREZ

Riobamba – Ecuador

2016

©2016, Dalila Lucía Toledo Sanmartín

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

El Tribunal de Tesis certifica que: El trabajo de investigación: ANÁLISIS DE LAS TECNOLOGIAS RDF, OWL, SWRL EN LA WEB SEMÁNTICA PARA CREAR ONTOLOGÍAS DE CURSOS VIRTUALES A PARTIR DE BASE DE DATOS DE UN LMS, de responsabilidad de la señorita Dalila Lucía Toledo Sanmartín, ha sido minuciosamente revisado por los Miembros de Tesis, quedando autorizada su presentación.

**Msc. Danilo Pastor**

**DIRECTOR DE TESIS**

\_\_\_\_\_

**Msc. Fernando Mejía**

**MIEMBRO DE TRIBUNAL**

\_\_\_\_\_

**Msc. Washington Luna**

**PRESIDENTE DE TRIBUNAL**

\_\_\_\_\_

**DOCUMENTALISTA**

**SISBIB ESPOCH**

\_\_\_\_\_

**NOTA DE LA TESIS:** \_\_\_\_\_

Yo, Dalila Lucía Toledo Sanmartín soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica de Chimborazo.

## **DEDICATORIA**

Dedico este trabajo principalmente a mi hija Dalila Arlett Buñay Toledo que es la bendición más grande que Dios me ha dado, ella ha sido un motivo más para salir adelante en las metas que me he propuesto, de igual manera dedico este trabajo a mis padres que fueron quienes me impulsaron día a día con sus consejos, a seguir adelante para culminar mi carrera a pesar de las dificultades que se presentaron siempre estuvieron presentes brindándome su apoyo incondicional, otra persona muy importante a quien quiero dedicar este trabajo es a mi abuelita mami Leti, que aunque ya no está entre nosotros físicamente, siempre estará presente en mi corazón, le dedico este trabajo ya que fue un pilar fundamental en mi vida, ella con sus consejos supo inculcarme la humildad y sobre todo la honestidad.

Dalila T.

## **AGRADECIMIENTO**

Agradezco a primeramente a Dios por darme la fuerza de voluntad y la salud para continuar con mi carrera, también agradezco a mis maestros que estuvieron ahí brindándome toda su ayuda para poder completar esta etapa de mi vida, en especial agradezco al Ing. Danilo Pastor quien ha sido más que un maestro un amigo que me ha brindado todo su apoyo en los momentos que más lo necesitaba.

Dalila T.

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS .....	xii
ÍNDICE DE FIGURAS.....	xiv
RESUMEN.....	xvi
SUMARY.....	xvii
INTRODUCCIÓN .....	1
CAPÍTULO I	
1. MARCO REFERENCIAL .....	2
1.1 Introducción.....	2
1.2 Antecedentes.....	2
1.3 Justificación del proyecto de tesis .....	4
1.3.1 Justificación teórica .....	4
1.3.2 Justificación práctica .....	5
1.4 Objetivos.....	6
1.4.1 Objetivo general .....	6
1.4.2 Objetivos específicos .....	6
1.5 Hipótesis .....	6
1.6 Métodos y técnicas .....	6
1.6.1 Métodos .....	6
1.6.2 Técnicas .....	7
CAPÍTULO II	
2. MARCO TEORICO.....	8
2.1 Introducción.....	8
2.2 Web semántica.....	8
2.3 Arquitectura de la web semántica.....	8

2.4	Ontologías de la web semántica .....	10
2.5	Áreas de aplicación de las ontologías.....	11
2.6	Componentes de una ontología.....	11
2.7	Tecnologías relacionadas con la web semántica.....	12
2.7.1	XML .....	12
2.7.2	HTML.....	13
2.8	HTML frente a XML.....	14
2.9	OWL.....	14
2.9.1	Sublenguajes de OWL.....	15
2.9.2	Capacidades de OWL.....	17
2.9.3	Elementos de OWL .....	17
2.10	RDF .....	19
2.10.1	Características de RDF .....	19
2.10.2	Formato de RDF .....	20
2.10.2.1	Modelo de datos .....	20
2.10.2.2	Sintáxis RDF.....	21
2.10.2.3	EsquemaRDF .....	21
2.10.2.4	Contenedores en RDF .....	22
2.11	SWRL .....	22
2.11.1	Características de SWRL.....	22
2.11.2	Reglas de SWRL .....	23
2.11.3	Especificación del comportamiento de gestión .....	24
2.12	LMS Moodle.....	25
2.12.1	Características.....	25
2.12.2	Ventajas de LMS Moodle.....	26
2.12.3	Desventajas de LMS Moodle.....	27



2.12.4	Estructura de LMS Moodle .....	27
2.12.4.1	Categorías.....	28
2.12.4.2	Cursos.....	28
2.12.4.3	Semanas y Temas .....	28
 CAPÍTULO III		
3.	ANÁLISIS DE TECNOLOGÍAS RDF, OWL Y SWRL.....	30
3.1	Introducción.....	30
3.2	Determinación de índices y parámetros de análisis .....	30
3.2.1	Descripción de indicadores y parámetros .....	31
3.2.2	Métodos para la evaluación de resultados .....	32
3.2.3	Valoración de Parámetros.....	33
3.3	Análisis de los índices y parámetros de comparación .....	34
3.3.1	Indicador 1: Productividad .....	34
3.3.1.1	Parámetro Tiempo de Procesamiento.....	34
3.3.1.2	Parámetro Consumo de Recursos.....	36
3.3.1.3	Resultados del Indicador de Productividad.....	37
3.3.2	Indicador 2: Usabilidad .....	40
3.3.2.1	Parámetro Facilidad de Aprendizaje .....	40
3.3.2.2	Parámetro Interoperabilidad.....	42
3.3.2.3	Resultados del Indicador de Usabilidad.....	43
3.3.3	Indicador 3: Procesamiento .....	45
3.3.3.1	Parámetro Acceso a datos .....	46
3.3.3.2	Parámetro Asociación de Información.....	47
3.3.3.3	Resultados del Indicador de Procesamiento.....	49
3.3.4	Indicador 4: Programación .....	51
3.3.4.1	Parámetro Familiaridad con otros Lenguajes.....	51

3.3.4.2	Parámetro Tiempo de Desarrollo .....	53
3.3.4.3	Resultados del Indicador Programación.....	54
3.4	Resumen del análisis .....	56
3.4.1	Fórmulas usadas para el análisis.....	56
3.4.2	Aplicación de fórmulas.....	57
3.4.3	Interpretacion de resultados.....	59
3.5	Comprobación de la hipótesis.....	59

#### CAPÍTULO IV

4.	DESARROLLO DE SOFTWARE PARA CONSTRUIR LA ONTOLOGÍA DE CURSOS VIRTUALES .....	61
4.1	Introducción.....	61
4.2	Análisis .....	61
4.2.1	Instalación de herramientas .....	61
4.2.1.1	Instalación de CentOS 6.5.....	62
4.2.1.2	Instalación de servidor web.....	62
4.2.1.3	Instalación de MySQL .....	63
4.2.1.4	Instalación de PHP .....	66
4.2.1.5	Instalación de Moodle 2.6.....	66
4.2.1.6	Instalación de phpmyadmin .....	71
4.2.1.7	Instalación de ARC2 .....	72
4.3	Metodología de desarrollo .....	72
4.3.1	Desarrollo de la Metodología .....	73
4.3.1.1	Planificación.....	73
4.3.1.2	Diseño .....	80
4.3.1.3	Codificación .....	89
4.3.1.4	Pruebas .....	96

## CAPÍTULO V

5.	RESULTADOS FINALES .....	100
5.1	Introducción.....	100
5.2	Demostración de resultados.....	100
5.3	Conclusión de Resultados.....	102
	CONCLUSIONES .....	103
	RECOMENDACIONES .....	104
	GLOSARIO DE TERMINOS	
	BIBLIOGRAFÍA	
	ANEXOS	

## ÍNDICE DE TABLAS

Tabla 1-3. Índices y parámetros de comparación.....	30
Tabla 2-3. Valoración de parámetros .....	33
Tabla 3-3. Criterios de evaluación del parámetro de Tiempo de Procesamiento.....	34
Tabla 4-3. Simbolización de Tiempo de Procesamiento.....	35
Tabla 5-3. Datos obtenidos del Parámetro Tiempo de Procesamiento.....	35
Tabla 6-3. Criterios de evaluación del parámetro de Consumo de Recursos.....	36
Tabla 7-3. Simbolización de Consumo de Recursos.....	36
Tabla 8-3. Datos obtenidos del Parámetro Consumo de Recursos.....	37
Tabla 9-3. Criterios de Evaluación de la Productividad.....	38
Tabla 10-3. Resultados obtenidos del Índice Productividad .....	38
Tabla 11-3. Criterios de evaluación para la Facilidad de Aprendizaje .....	40
Tabla 12-3. Simbolización de Facilidad de Aprendizaje .....	41
Tabla 13-3. Datos obtenidos del Parámetro Facilidad de Aprendizaje .....	41
Tabla 14-3. Criterios de evaluación del parámetro de Interoperabilidad.....	42
Tabla 15-3. Simbolización de Interoperabilidad .....	42
Tabla 16-3. Datos obtenidos del Parámetro Interoperabilidad.....	43
Tabla 17-3. Criterios de Evaluación de la usabilidad.....	43
Tabla 18-3. Resultados obtenidos del Índice Usabilidad .....	44
Tabla 19-3. Criterios de evaluación parámetro Acceso a Datos .....	46
Tabla 20-3. Simbolización de Acceso a Datos.....	46
Tabla 21-3. Datos obtenidos del Parámetro Acceso a Datos .....	47
Tabla 22-3. Criterios de evaluación del parámetro de Asociación de Información .....	47
Tabla 23-3. Simbolización de Asociación de Información .....	48
Tabla 24-3. Datos obtenidos del Parámetro Asociación de Información.....	48

Tabla 25-3. Criterios de Evaluación del Procesamiento .....	49
Tabla 26-3. Resultados obtenidos del Índice Procesamiento .....	49
Tabla 27-3. Criterios de evaluación parámetro Familiaridad con otros Lenguajes.....	51
Tabla 28-3. Simbolización de Familiaridad con otros Lenguajes .....	52
Tabla 29-3. Datos obtenidos del Parámetro Familiaridad con otros Lenguajes.....	52
Tabla 30-3. Criterios de evaluación del parámetro Tiempo de Desarrollo .....	53
Tabla 31-3. Datos obtenidos del Parámetro Tiempo de desarrollo .....	53
Tabla 32-3. Criterios de Evaluación de la Programación.....	54
Tabla 33-3. Resultados obtenidos del Parámetro Programación.....	54
Tabla 34-3. Resultados Totales del Análisis Comparativo .....	58

## ÍNDICE DE FIGURAS

Figura 1-2. Componentes de una ontología .....	12
Figura 2-2. Sentencia de RDF.....	21
Figura 3-2. Sintaxis de una regla en SWRL.....	23
Figura 4-2. Arquitectura del gestor semántico de SWRL .....	25
Figura 5-2. Esquema de Moodle .....	27
Figura 1-3. Resultado del Índice Productividad.....	39
Figura 2-3. Porcentajes obtenidos de índice Productividad .....	39
Figura 3-3. Resultado del Índice Usabilidad.....	44
Figura 4-3. Porcentajes obtenidos del índice Usabilidad .....	45
Figura 5-3. Resultado del Índice Procesamiento.....	50
Figura 6-3. Porcentajes obtenidos del Índice Procesamiento.....	50
Figura 7-3. Resultado del índice Programación .....	55
Figura 8-3. Porcentajes obtenidos del índice Programación .....	55
Figura 9-3. Resultado del análisis .....	58
Figura 10-3. Porcentajes Obtenidos del Análisis .....	59
Figura 1-4. Instalación de HTTP.....	62
Figura 2-4. Inicio de Servicio Web.....	63
Figura 3-4. Inicio Automático de HTTP .....	63
Figura 4-4. Instalación de MySQL.....	63
Figura 5-4. Inicio de Servicio MySQL .....	64
Figura 6-4. Configuración de Seguridades de MySQL.....	64
Figura 7-4. Seguridades MySQL .....	65
Figura 8-4. Inicio Automático de MySQL.....	65
Figura 9-4. Instalación de PHP .....	66

Figura 10-4. Descarga del paquete de Moodle.....	66
Figura 11-4. Descomprensión del Paquete de Moodle .....	67
Figura 12-4. Ingreso a la Base MySQL .....	67
Figura 13-4. Creación de la Base de Moodle .....	68
Figura 14-4. Asignación de privilegios a la base de Moodle. ....	68
Figura 15-4. Movimiento de los archivos Descomprimidos a la Base de Moodle. ....	68
Figura 16-4. Creación de la carpeta datos de Moodle.....	69
Figura 17-4. Configuración de base, nombre de usuario y password de Moodle. ....	70
Figura 18-4. Configuración de la IP del servidor de Moodle.....	70
Figura 19-4. Configuración de la localización de la carpeta data. ....	71
Figura 20-4. Fases de la Metodología XP .....	73
Figura 21-4. Esquema de la base de Moodle .....	87
Figura 22-4. Diseño de aplicación para buscar un curso.....	88
Figura 23-4. Diseño aplicación que presenta la información de un curso .....	88
Figura 24-4. Diseño mientras se genera el archivo de la ontología .....	89
Figura 25-4. Ejemplo archivo .owl .....	95
Figura 1-5. Inicio de Aplicación .....	100
Figura 2-5. Búsqueda de Cursos .....	101
Figura 3-5. Presentación de Resultados .....	101
Figura 4-5. Comprobación de datos obtenidos.....	102

## RESUMEN

El trabajo realizado tuvo como objetivo estudiar la arquitectura de la web semántica y la estructura de las bases de datos de un LMS Moodle, así como analizar y determinar las tecnologías de la web semántica que permitan crear ontologías y extraer datos a partir de un entorno LMS libre. Se desarrolló un módulo de software que permite construir una ontología a partir de la información de base de datos del LMS Moodle y así realizar la validación de la extracción de los datos del LMS Moodle hacia la ontología estructurada en un entorno virtual de la Escuela Superior Politécnica de Chimborazo. Para el desarrollo de la aplicación que genera una ontología a partir de los datos de un curso almacenado en una base de Moodle, se trabajó con la metodología ágil XP que está orientada al desarrollo rápido de aplicaciones, y las interfaces que se desarrollan son simples e intuitivas para los usuarios. Posteriormente al análisis realizado se obtuvo como resultado que las tecnologías RDF y OWL permiten la estructuración de ontologías de forma más óptima extrayendo datos de las base de datos de un LMS Moodle, mientras que con la tecnología SWRL también se puede realizar, tomando en cuenta que se requiere de más recursos. Por lo tanto se pudo concluir que las tecnologías de la web semántica OWL y RDF, y la estructura de las bases de datos de un LMS Moodle combinadas entre sí, permiten generar ontologías y extraer datos desde un entorno LMS libre. Así como también que desarrollo de un módulo de software que permite construir una ontología a partir de la información de base de datos del LMS es posible, mediante el uso de herramientas como ARC y el lenguaje de programación php. Este módulo es relativamente complejo ya que su estructura depende de la ontología construida. Y se recomienda que se debe analizar otras alternativas de librerías diferentes a ARC para trabajar con PHP en el proceso de tecnología de web semántica.

**Palabras claves:** <TECNOLOGIA Y CIENCIAS DE LA INGENIERIA>, <INGENIERIA DE SISTEMAS>, <BASE DE DATOS>, <RDF>, <OWL>, <SWRL>, <LMS MOODLE>



## SUMMARY

The work aimed to study the architecture of the semantic web and the structure of the databases of a LMS Moodle, as well as analyze and determine the technologies of the Semantic Web that create ontologies and extract data from one LMS environment free. A software module that allows you to build an ontology from information database LMS Moodle and perform the validation of the extraction of data from LMS Moodle to ontology structured in a virtual environment of the Escuela Superior Politecnica de Chimborazo. For the development of the application that generates an ontology from the data of a course stored in a database Moodle, we worked with agile methodology XP that is oriented rapid application development, and interfaces that are developed are simple and intuitive for users. Following the analysis was obtained as a result that RDF and OWL technologies allow structuring ontologies more optimally extracting data from the database of a LMS Moodle, while technology SWRL also can be done, taking into account that it requires more resources. Therefore it was concluded that the technologies of the Semantic Web OWL and RDF, and the structure of the databases of a Moodle LMS in combination, can generate ontologies and extract data from a free LMS environment. As well as to develop a software module that allows you to build an ontology from information database LMS is possible, by using tools such as ARC and the php programming language. This module is relatively complex since its structure depends on the ontology built. And it is recommended that other alternatives must be analyzed from different libraries to ARC to work with PHP in the process of semantic web technology.

**KEYWORDS:** <TECHNOLOGY AND ENGINEERING SCIENCES>, <SYSTEMS ENGINEERING> <DATABASE>, <RDF>, <OWL>, <SWRL>, <LMS MOODLE>

## **INTRODUCCIÓN**

El presente trabajo de investigación se basa en la necesidad que tienen los usuarios para realizar búsquedas más concisas e inteligentes, en este caso se orientó principalmente a los contenidos de los diferentes cursos virtuales que se encuentran almacenados en una base de datos de Moodle. Para mejorar la experiencia de búsquedas se requiere de una web semántica la cual mediante sus tecnologías permite estructurar ontologías las mismas que se generan a partir de la información extraída desde Moodle. La web semántica permite realizar búsquedas mediante conceptos mas no mediante la comparación de cadenas de caracteres, es decir esta hace posible sostener una interacción entre un usuario y un agente de software. Una Ontología se trata de una jerarquía de conceptos, con atributos y relaciones, las cuales proporcionan de un vocabulario consensuado para definir redes semánticas de unidades de información interrelacionadas. Una aplicación informática ayudaría mucho en el sentido del tiempo que lleva realizar actividades de búsqueda así como también se obtendría mayor provecho de los recursos físicos y lógicos.

El documento se encuentra estructurado de tal manera que detalla cada una de las etapas del proceso de desarrollo investigativo realizado en cinco capítulos, los cuales están organizados de la siguiente manera. En el primer capítulo se encuentra detallada la justificación, los objetivos, los métodos y técnicas aplicadas así como también la hipótesis planteada para la realización del trabajo, dicha hipótesis ha sido considerada como aceptada ya que las metodologías de la web semántica si permiten estructurar ontologías a partir de los datos extraídos de un LMS. El segundo capítulo está orientado principalmente a la parte teórica, aquí se encuentran los conceptos principales a cerca de lo que es la web semántica, la descripción de las tecnologías de web semántica OWL, RDF y SWRL, así como también la descripción de Moodle. En el tercer capítulo se encuentra detallado el análisis realizado a cerca de cada una de las tecnologías estudiadas, para el análisis se realizó la definición de los indicadores y sus respectivos parámetros los cuales fueron definidos en base a la información que se encuentra en el capítulo 2. En el capítulo esta detallada la metodología aplicada para el desarrollo de la aplicación informática, la cual permite obtener la información de los cursos virtuales almacenados en la base de Moodle. Para concluir se definió el quinto capítulo en el cual se describe el funcionamiento de la aplicación desarrollada y se demuestra la validación de los datos extraídos mediante la aplicación, en relación a los datos almacenados en la base de Moodle.

# CAPÍTULO I

## 1. MARCO REFERENCIAL

### 1.1 Introducción

En este capítulo se describen los diferentes aspectos por los cuales se ha procedido a realizar la investigación y análisis de las diferentes tecnologías para la web semántica. Se describen los antecedentes en los cuales se describe el principal problema encontrado, tomándolo como base para realizar el estudio, la aplicación que se va a dar y los objetivos planteados para cumplir durante la investigación.

### 1.2 Antecedentes

Los sitios y páginas web no poseen la capacidad de indizar los motores de búsqueda de uso público, como por ejemplo Google o Alta vista. Estos fueron creados originalmente para descargar, leer e indizar páginas HTML, y cualquier otro formato era ilegible, es decir invisible para dichos motores de búsqueda.

Podemos acceder a las páginas principales de los sitios web que proporcionan acceso a bases de datos, porque tales principales son páginas HTML convencionales, pero no se puede acceder al resto del sitio a través del motor de búsqueda; ya que el resto del sitio puede ser una enorme base de datos. (CODINA, 2003 pág. 5)

El problema por el cual se pretende realizar este estudio se refiere principalmente a que los actuales buscadores no tienen la capacidad de ofrecer búsquedas con tasas de precisión mínimamente adecuadas en sus resultados, evidenciando de esta manera que las técnicas de búsquedas estadísticas no pueden solucionar por sí solas el problema de recuperar la información. Por lo tanto la idea es pasar de una web de enlace de documentos a una web de conceptos, destacando de esta manera la relación entre conceptos y más no de los vínculos o hipervínculos, dándole al computador la posibilidad de hacer búsquedas con precisión.

Cuando alguien realiza una consulta en un motor de búsqueda, lo que hace dicho motor es comparar las palabras de la pregunta con las palabras de su índice. Si encuentra un documento que tenga la consulta realizada, lo devolverá como respuesta. “Con estas limitaciones, aunque la búsqueda en internet está repleta de satisfacciones, también provoca mucha frustración, ya que la información obtenida será limitada, debido a la incapacidad de los motores de búsqueda para diferenciar y precisar puntos de vista por ejemplo, si los resultados obtenidos de una consulta son artículos de revistas o entradas de diccionario.” (ROVIRA, 2006 p. 3)

La web semántica está dotada de mayor significado en donde el usuario de internet pueda encontrar respuestas a sus preguntas. Esta web está basada en una infraestructura de metadatos la cual aporta un camino para razonar en la web extendiendo sus capacidades. La web semántica utiliza varias tecnologías como RDF (Resource Descriptor Framework), OWL (Language Web Ontology) y SWRL (Semantic Web Rule Language) las mismas que ayudan a convertir la web en una infraestructura global en la que es posible compartir y reutilizar datos y documentos entre diferentes tipos de usuarios. Cada una de estas tecnologías se van complementando una con la otra, es decir que RDF es la tecnología base de la web semántica ya que proporciona información descriptiva de los recursos de la web, utiliza las URI (Uniform Resource Identifier) para nombrar la relación entre los dos extremos de una conexión, esto permite que los datos estructurados y semi-estructurados sean expuestos y compartidos a través de la diferentes aplicaciones, (W3C, 2004 p. 1). Esta tecnología se complementa con OWL que es la tecnología que me permite crear una ontología, esta tecnología puede ser usada para presentar explícitamente el significado de términos en vocabularios y las relaciones de entre dichos términos (KSARMIENTO, 2009 p. 6), esta tecnología a su vez esta se complementa con SWRL que es la que me va a permitir hacer relaciones entre conceptos obtenidos mediante RDF. Este es un lenguaje que se extiende OWL incluyendo Horn-like rules, que se pueden combinar con la base de conocimiento de OWL. Ofrece una sintaxis abstracta de alto nivel que amplía la de OWL (LAGE, 2011 p. 3).

Los LMS, por su parte han tenido gran acogida sobre todo en los niveles de educación superior en donde la educación en base a libros físicos, trabajos impresos y tareas diarias poco a poco han ido desapareciendo dando lugar a la utilización de archivos informáticos los mismos que son administrados de manera mucho más ordenada dentro de un aula virtual. Por ende se requiere de una herramienta que me permita mejorar la utilización de los recursos de un LMS disponibles pero no se sabe que tecnología de web semántica se puede usar para crear cursos virtuales que posean un lenguaje procesable y entendible no solo por los humanos sino también por los ordenadores.

Debido a los profundos cambios a los que se ha visto sometida la web, a nivel de comunicaciones, negocios y en especial a la educación realizaremos un estudio para modelar un curso virtual a través de ontologías para que se pueda compartir información de forma estandarizada.

### **1.3 Justificación del proyecto de tesis**

#### ***1.3.1 Justificación teórica***

La web semántica permite realizar búsquedas mediante conceptos mas no mediante la comparación de cadenas de caracteres, es decir esta hace posible que sostener una interacción entre un usuario y un agente de software. (ROVIRA, 2006 p. 8). Una Ontología se trata de una jerarquía de conceptos, con atributos y relaciones, las cuales proporcionan de un vocabulario consensuado para definir redes semánticas de unidades de información interrelacionadas. (CASTELLS P, 2011 p. 4).

Para poder explotar la web semántica, se necesitan lenguajes de marcado apropiados que representen el conocimiento de las ontologías.

Los niveles de representación necesitados para los modelos de ontologías son:

- Sintaxis
- Estructura
- Semántica
- Pragmática.

Ontología proviene de la filosofía; pero en Inteligencia Artificial, tiene diferentes connotaciones. La definición propuesta por Gruber y extendida por Studer y sus colegas describe la ontología como “Una especificación explícita y formal sobre una conceptualización compartida”.<sup>1</sup>

Es decir que la ontología define conceptos y relaciones de algún dominio de forma compartida y consensuada, la misma que puede ser representada de una manera formal legible y utilizable por los ordenadores. (TELLO, 2010 p. 10).

---

<sup>1</sup> Resumiendo el concepto dado diremos que una ontología son métodos y conceptos organizados de tal forma que puedan ser aplicados en la web semántica.

Por lo tanto es importante realizar un análisis de las tecnologías RDF, OWL y SWRL de la web semántica para determinar cuál de ellas es la que me da la pauta idónea para aplicarla en los cursos virtuales que son herramientas que se están utilizando muy ampliamente en la educación, sobre todo en el nivel superior. Es así que el presente proyecto está planteado de tal manera que al momento de crear una ontología esta nos brinde la opción de compartir, reutilizar e intercambiar la información que existe en un curso, para de esta manera optimizar el uso de recursos tanto informáticos como pedagógicos.

### ***1.3.2 Justificación práctica***

Para la realización del proyecto planteado se pretende crear un módulo de un curso virtual en Moodle, el mismo que nos servirá como entorno de pruebas para la aplicación de la ontología. La ontología por su parte nos va a permitir obtener información de las bases de datos de los LMS, dicha información será procesada de acuerdo a las necesidades requeridas, por ejemplo si necesitamos crear un curso virtual para una materia específica, actualmente lo que se realiza es la creación de un curso sin ningún contenido y poco a poco se van implementando nuevas tareas y recursos de acuerdo a las necesidades del usuario, lo cual se vuelve complicado, por lo que si tenemos una ontología que abstraiga la información de un curso ya existente esta actividad será mucho más rápida y concisa, ya que la ontología dará la opción de reutilizar y optimizar los recursos existentes de un curso virtual. Esto ayudaría mucho en el sentido del tiempo que lleva realizar este tipo de actividades así como también se obtendría mayor provecho de los recursos físicos y lógicos.

Es decir el desarrollo de actividades de enseñanza y aprendizaje, y el conocimiento almacenado en las ontologías, permitirá que las aplicaciones puedan extraer automáticamente datos de las bases de datos, procesarlos y sacar conclusiones de ellos, así como tomar decisiones y negociar con otros agentes o personas. La evaluación de la ontología será dentro de un ambiente de pruebas controlado. Para lo cual se basará en el módulo que será creado para realizar las pruebas respectivas, y debido a que el sistema a crearse (dentro del Moodle) debe leer las tablas relacionadas a ese curso e ir extrayendo hacia la Ontología, la misma que leerá cuantos foros, cuantas tareas, etc. existen.

Es decir que sirve para extraer de un determinado curso (el que especifiquemos) y para evaluar su funcionalidad ya en un entorno real esto se lo realizará en el e-virtual de la ESPOCH, ya que si

funciona en el entorno de pruebas funcionará para cualquier otro curso porque las tablas de la BD son las mismas.

## **1.4 Objetivos**

### ***1.4.1 Objetivo general***

Realizar un análisis de las tecnologías RDF, OWL, SWRL en la web semántica para crear ontologías de cursos virtuales a partir de base de datos de un LMS.

### ***1.4.2 Objetivos específicos***

- Estudiar la arquitectura de la web semántica y la estructura de las bases de datos de un LMS Moodle.
- Analizar y determinar las tecnologías de la web semántica que permitan crear ontologías y extraer datos a partir de un entorno LMS libre.
- Desarrollar un módulo de software que permita construir una ontología a partir de la información de base de datos del LMS.
- Realizar la validación de la extracción de los datos del LMS Moodle hacia la ontología estructurada en un entorno virtual de la ESPOCH.

## **1.5 Hipótesis**

El uso de las tecnologías RDF, OWL y SWRL, permiten estructurar de forma semántica los cursos virtuales a partir de los datos extraídos de la base de datos de un LMS.

## **1.6 Métodos y técnicas**

### ***1.6.1 Métodos***

Para el análisis de las tecnologías RDF, OWL, SWRL en la web semántica para crear ontologías de cursos virtuales a partir de base de datos de un LMS se aplicará el método científico el cual consta de las siguientes actividades.

- Planteamiento del problema.
- Formulación de la Hipótesis.
- Levantamiento de la información.
- Análisis e interpretación de Resultados.
- Comprobación de hipótesis.
- Difusión de resultados.

Para el desarrollo del prototipo de ontologías de cursos virtuales se utilizará la metodología híbrida, desarrollada para procesos de diseño de aplicaciones Web, denominada también “Metodología para Creación de Sitios Web”, la misma que fue creada por el Ingeniero Dely Maybel Gil Álvarez; consta de las siguientes fases:

- Análisis
- Planificación.
- Contenido.
- Diseño.
- Programación.
- Testeo.
- Mercadeo y Publicidad.

### ***1.6.2 Técnicas***

Las técnicas que utilizaremos para la recopilación de información serán la observación y la revisión de documentos. Para el análisis de las tecnologías de la web semántica usaremos también la observación, generación de pruebas y la aplicación de actividades en el prototipo para evaluarlas mediante una encuesta, esto nos ayudará a conocer si el estudio realizado cumple con las expectativas planteadas, en cuanto a la usabilidad de la web semántica aplicada en la creación de cursos virtuales.



## **CAPÍTULO II**

### **2. MARCO TEORICO**

#### **2.1 Introducción**

En este capítulo se describe todo lo investigado acerca de las tecnologías de la web semántica como, conceptos, evolución, clasificación y arquitectura, para poder ejecutar correctamente una aplicación en el campo de la educación. El avance de la tecnología diariamente genera nuevas herramientas para facilitar la experiencia de los usuarios en la web, por ello se está avanzando un paso más al crear una web inteligente que sea capaz de interpretar y dar sentido a una frase, para de esta manera agilizar las actividades de búsqueda. Por lo tanto podría decir que es una tecnología nueva y esto implica que no se cuenta con muchos recursos bibliográficos como por ejemplo libros o folletos, por tal motivo para abordar todo lo inmerso en esta tecnología se utilizaron recursos de investigación digitales como páginas web, Artículos Científicos o PAPERS<sup>3</sup>, tesis realizadas, memorias publicadas, blogs, etc.

#### **2.2 Web semántica**

La web semántica es un avance de la web tradicional, ya que esta permite realizar búsquedas precisas, mediante el significado de las palabras y los recursos existentes en la web. Es decir al momento que utilizamos un navegador para realizar una búsqueda, la semántica nos ayudara a dar sentido a la oración ingresada. El uso de la semántica en la tecnología permitirá enlazar y dar sentido a cada una de las palabras dependiendo su ubicación. La web semántica permite agilizar las búsquedas por lo tanto de esta manera optimizamos el tiempo de navegación y búsqueda. (CEJUDO, 2011 pág. 19)

#### **2.3 Arquitectura de la web semántica**

La web semántica consta de diferentes componentes para su funcionamiento, los mismos que cumplen una función específica dentro de la web 3.0 o más conocida como web semántica. Estos componentes SON (LAPUENTE, 2013 pág. 6):

- XML: Provee una sintaxis elemental para las estructuras de recursos dentro de documentos.
- XML Schema: Este es el lenguaje que facilita y restringe la estructura y el contenido de los recursos almacenados dentro de documentos XML.
- RDF: Es el lenguaje que sirve para expresar modelos de los datos, que se refieren a los objetos o también llamados recursos, y a sus relaciones. Esto se puede representar en sintaxis de XML.
- RDF Schema: Es un vocabulario para describir propiedades y clases de recursos basados en RDF con semántica, para extender jerarquías de las propiedades y clases.
- OWL: Es similar a una metodología para desarrollar temas o vocabularios específicos en los que podamos asociar esos recursos.

Para una mejor comprensión se puede decir que la web semántica está estructurada en diferentes capas, las mismas que son:

**Capa Unicode:** Esta capa es estándar, su objetivo es proporcionar el medio por el cual un texto en cualquier forma e idioma pueda ser codificado para el uso informático. El mismo nos permite mostrar información en cualquier idioma y con la certeza de que no aparezcan símbolos extraños.

**Capa de las URI:** En esta capa se encuentran cadenas, las mismas que permiten acceder a cualquier recurso de la Web. En la Web Semántica la función de las URIs es identificar los objetos o recursos. Si dos objetos cuentan con la misma URI pueden existir problemas.

**XML+NS+xmlschema:** Esta es la capa más técnica de la Web Semántica aquí se encuentran agrupadas las diferentes tecnologías que posibilitan la comunicación entre agentes. El lenguaje XML nos ofrece un formato común para el intercambio de documentos Namespaces (NS) proporciona un método para cualificar elementos y atributos de nombres usados en documentos XML asociándolos con espacios de nombre identificados por referencias a URIs. XML Schema es un lenguaje que permite describir la estructura y restringir el contenido de documentos XML.

**RDF+rdfschema:** Esta capa se basa en la capa anterior, define el lenguaje universal con el que podemos expresar diferentes ideas en la Web Semántica. RDF es un lenguaje que define un modelo de datos para describir recursos mediante tríos sujeto-predicado-objeto.

Los dos primeros serán URIs y el tercero puede ser URI o un valor literal. RDF Schema es un vocabulario RDF que nos permite describir recursos mediante una orientación a objetos. Esta capa no sólo ofrece una descripción de los datos, sino también cierta información semántica como:

**Ontologías:** Nos permiten clasificar la información. Esta capa permite extender la funcionalidad de la Web Semántica agregando nuevas clases y propiedades para describir los recursos.

**Lógica:** Además de ontologías se precisan reglas de inferencia.

**Pruebas:** Se intercambiarán “pruebas” escritas en el lenguaje unificador de la Web Semántica. Este lenguaje posibilita las inferencias lógicas realizadas a través del uso de reglas de inferencia.

**Confianza:** Hasta que no se haya comprobado de forma exhaustiva las fuentes de información, los agentes deberían ser muy prevenidos acerca de lo que leen en la Web Semántica.

**Firma digital:** Utilizada por los ordenadores y agentes para verificar que la información ha sido ofrecida por una fuente de confianza.

## 2.4 Ontologías de la web semántica

Se trata de una jerarquía de conceptos, con atributos y relaciones, las cuales proporcionan de un vocabulario consensuado para definir redes semánticas de unidades de información interrelacionadas. (CASTELLS P, 2011 p. 7). Ontología proviene de la filosofía; pero en IA, tiene diferentes connotaciones. La definición propuesta por Gruber y extendida por Studer y sus colegas describe la ontología como “Una especificación explícita y formal sobre una conceptualización compartida”.<sup>2</sup>

---

<sup>2</sup> Resumiendo el concepto dado diremos que una ontología son métodos y conceptos organizados de tal forma que puedan ser aplicados en la web semántica.

Es decir que la ontología define conceptos y relaciones de algún dominio de forma compartida y consensuada, la misma que puede ser representada de una manera formal legible y utilizable por los ordenadores. (TELLO, 2010 p. 3).

## **2.5 Áreas de aplicación de las ontologías**

Las ontologías pueden ser aplicadas en distintos campos tanto educativos, de información y comerciales. En lo educativo las ontologías pueden ayudar a la creación de agentes especializados en un tema por ejemplo en las aulas virtuales.

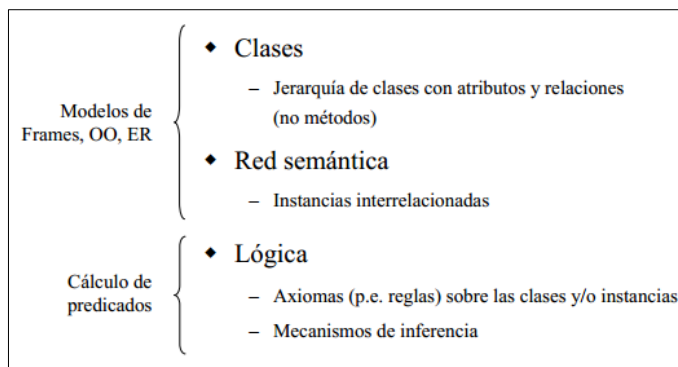
En el área de información también son muy utilizadas ya que pueden ser aplicadas conjuntamente con la web semántica para la realización de búsquedas. Y finalmente en lo económico es muy utilizado ya que al tener una web semántica que permita realizar búsquedas concretas, se puede mejorar el comercio electrónico. (CASTELLS, 2003 p. 9).

En concreto podemos decir que las áreas en las que se pueden aplicar las ontologías son:

- Ingeniería del conocimiento
- Gestión del conocimiento
- Procesamiento del lenguaje natural
- Sistemas de información cooperativos
- Integración inteligente de información
- Recuperación de información
- Comercio electrónico

## **2.6 Componentes de una ontología**

Los principales componentes de una ontología son los modelos de las tramas dentro de las cuales tenemos las clases y la red semántica, y el cálculo de predicados que es con lo cual la web semántica va a poder entender una búsqueda. Las clases de las ontologías en sí como toda clase dentro de programación, tiene sus atributos y relaciones, pero en las ontologías los métodos están fuera de la clase. Mientras que en la web semántica se van a especificar las instancias que se van a interrelacionar. (PABLO DIEZ SUAREZ, 2008 p. 14)



**Figura 1-2.** Componentes de una ontología

Realizado por: Dalila Toledo

## 2.7 Tecnologías relacionadas con la web semántica

La web semántica requiere del uso de varias tecnologías para su implementación, las mismas que cumplen un rol específico en cada una de las capas que conforman la web semántica. Por ejemplo XML es una extensión avanzada de HTML, el mismo que es el lenguaje principal de las aplicaciones web.

### 2.7.1 XML

XML es el Lenguaje de Etiquetado Extensible muy simple pero estricto el mismo que tiene un papel muy importante en el intercambio de una gran variedad de datos. Es muy similar a HTML pero su principal función es describir los datos mas no mostrarlos como en el caso de HTML. XML en si es un formato que permite la lectura de datos a través de diferentes aplicaciones (Consortion, 2014 pág. 18).

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

XML utiliza un conjunto de etiquetas para definir elementos de datos. Cada elemento encapsula una parte de datos que puede ser muy simple o muy compleja. Puede definir un conjunto ilimitado de etiquetas XML. Por ejemplo, podría definir etiquetas XML para declarar partes de datos de un pedido, como el precio, los impuestos, la dirección de envío, la dirección de facturación, etc. A medida que se adopten etiquetas XML en toda la organización y a través de las organizaciones, los

datos de todos los tipos de almacenes de datos distintos serán más fáciles de intercambiar y manipular (Consortion, 2014 pág. 19).

XML es una simple plataforma independiente y un estándar muy utilizado. La ventaja de XML sobre HTML es que para la interfaz de usuario de los datos estructurados. Esta separación de datos de la aplicación habilita la integración de datos desde diversos orígenes. (Consortion, 2014 pág. 19).

### 2.7.2 *HTML*

Es el lenguaje en que se escriben los millones de documentos que hoy existen en el World Wide Web. Cuando accedemos a uno de estos documentos, el cliente o explorador los interpreta y los muestra.

Los documentos HTML son ficheros de texto plano (también conocidos como ASCII) que pueden ser editados con cualquier editor de texto (como Vim o el "Bloc de notas" de Windows). También podríamos utilizar cualquier programa procesador de textos (como StarWriter o Word), salvando el documento como "solo texto". El nombre de los ficheros escritos en lenguaje HTML suelen tener la extensión html o htm (GAVIOTA, 2014 pág. 6).

El lenguaje HTML se basa en la sintaxis SGML (Standard Generalized Markup Language). Esto quiere decir que los diferentes elementos (párrafos, encabezamientos, tablas, listas, etc.) de un documento para la WWW se señalan intercalando etiquetas que indican al navegador cómo debe mostrarlo (RUIZ, 2008 pág. 4).

Una etiqueta HTML consiste en un signo menor "<", un nombre de una directiva (orden o comando para el navegador), seguido de los parámetros o atributos y un signo mayor ">". Para cualquier etiqueta que indica el inicio de un elemento hay otra de cierre que indica que esa directiva ya no debe actuar sobre el texto que sigue (RUIZ, 2008 pág. 5).

`<directiva parámetro="xxxx"> ... </directiva>`

HTML no es sensible a mayúsculas y minúsculas. Para HTML es equivalente <HTML> y <html>, con algunas excepciones. Aunque es recomendable es escribir los nombres de las etiquetas en minúsculas ya que las nuevas generaciones del HTML están basadas en SGML que requiere de

documentos bien formados (GAVIOTA, 2014 pág. 3). Así que, si nuestra intención es continuar trabajando con este tipo de documentos en el futuro, nos puede venir bien seguir el modo de escribir de los documentos bien formados:

- Nombres de etiquetas y atributos en minúsculas.
- Etiquetas de cierre para elementos no vacíos.
- Los valores de los atributos deben estar incluidos entre comillas dobles.

## **2.8 HTML frente a XML**

A continuación se presentan algunas de las características diferenciales entre HTML y XML, que son las tecnologías en las cuales se basa la web semántica. (RUIZ, 2008 pág. 8)

- XML no es un sustituto de HTML; de hecho, XML se puede considerar como un complemento de HTML. XML y HTML tienen objetivos distintos: HTML está diseñado para mostrar datos y se centra en la apariencia de los datos, mientras que XML está diseñado para describir datos y está centrado en lo que son los datos.
- Al igual que HTML, XML no hace nada más. Mientras que es posible utilizar las etiquetas XML para describir la estructura de un elemento como un pedido, este lenguaje no contiene ningún código que se pueda utilizar para enviar ese pedido, procesarlo o asegurarse de que está completo. Otras personas deben escribir código para hacer realmente estas cosas con sus datos en formato XML.
- A diferencia del lenguaje HTML, el autor de un esquema o documento define las etiquetas XML y estas son ilimitadas. Las etiquetas HTML están predefinidas; los autores de HTML sólo pueden utilizar etiquetas compatibles con el estándar HTML actual.

## **2.9 OWL**

OWL está pensado para ser usado cuando la información contenida en los documentos necesita ser procesada por las aplicaciones, al contrario que en las situaciones donde el contenido sólo necesita ser presentado a los humanos. OWL puede ser usado para representar explícitamente el significado de términos en vocabularios y las relaciones entre esos términos (KSARMIENTO, 2009 p. 5).

Esta representación de términos y sus interrelaciones se denomina ontología. OWL tiene mayor capacidad para expresar significado y semántica que XML, RDF, y RDF-S, y, de este modo, OWL va más allá de estos lenguajes en su capacidad para representar contenido interpretable por un ordenador en la Web (PABLO DIEZ SUAREZ, 2008 p. 4).

OWL es similar, pero más grande, mejor y más malo. OWL te permite decir mucho más acerca de su modelo de datos, le muestra cómo trabajar de manera eficiente con las consultas de bases de datos y razonadores automáticas, y proporciona útiles anotaciones para llevar sus modelos de datos en el mundo real. OWL no sólo indica cómo puede usar cierto vocabulario, en realidad indica cómo no puede usarlo. En OWL, por lo menos en algunas instancias de OWL, existen afirmaciones que no son realmente legales, simplemente no se permite decir que algo puede ser tanto una clase y una instancia a la vez. OWL permite decidir qué tan expresivo desea ser, dadas las realidades computacionales involucradas. De hecho, OWL permite restringir las opciones de modelado de datos a las que permitan dar consultas más rápidas de búsqueda, los mismo que permiten el razonamiento conceptual, o aquellos que pueden ser fácilmente implementados con motores de reglas (SEMANTICA, 2015 p. 3).

Por ejemplo, OWL incluye todos sus viejos amigos de RDF tales como `rdfs: type`, `rdfs: domain`, y `rdfs: subPropertyOf`. Sin embargo, por ejemplo, OWL permite describir datos en términos de operaciones de conjuntos como:

*Example:Mother owl:unionOf (Example:Parent, Example:Woman)*

Permite definir las equivalencias entre bases de datos:

*AcmeCompany:JohnSmith owl:sameAs PersonalDatabase:JohnQSmith*

Se le permite restringir el valor de las propiedades:

*Example:MyState owl:allValuesFrom (State:NewYork, State:California, ...)*

Owl proporciona el vocabulario tanto nuevo como sofisticado para su uso en el modelado de datos y el razonamiento que obtiene su propia lección (SEMANTICA, 2015 p. 4).

### **2.9.1 Sublenguajes de OWL**

OWL proporciona tres lenguajes, cada uno con nivel de expresividad mayor que el anterior, diseñados para ser usados por comunidades específicas de desarrolladores y usuarios.



**OWL Lite:** Está diseñado para aquellos usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples. Por ejemplo, a la vez que admite restricciones de cardinalidad, sólo permite establecer valores cardinales de 0 o 1 (PABLO DIEZ SUAREZ, 2008 p. 7).

Facilita la compatibilidad con otros modelos o paradigmas, así como el desarrollo de herramientas de autor. Tiene un razonamiento eficiente, proporciona desarrollo de herramientas y compatibilidad con otros modelos. Es el menos expresivo, se usa en situaciones donde sólo existe una simple jerarquía de clases y donde se necesitan pocas restricciones.

**OWL DL:** Está diseñado para aquellos usuarios que requieren un término medio en cuanto a la expresividad, conservando completitud computacional y que todos los cálculos se resuelvan en un tiempo finito. OWL DL incluye todas las construcciones del lenguaje de OWL, pero sólo pueden ser usados bajo ciertas restricciones (PABLO DIEZ SUAREZ, 2008 p. 7).

Está basado en lógica descriptiva, la cual es una lógica basada en primer orden; es favorable para el razonamiento automático, para la clasificación de jerarquías y para detectar las inconsistencias en las ontologías. Limita la expresividad intentando conseguir decidibilidad y máxima expresividad manteniendo completitud y computabilidad. Usa restricciones para asegurar la existencia de un procedimiento de razonamiento decidable. No puede haber restricciones de cardinalidad (locales ni globales) en propiedades transitivas, ni sus inversas, ni sus súper propiedades. No pueden faltar definiciones. Los axiomas de desigualdad sólo se pueden aplicar a instancias con nombre (PABLO DÍEZ SUÁREZ, 2008 pág. 7)

**OWL Full:** Está dirigido a usuarios que quieren máxima expresividad y libertad sintáctica de RDF sin garantías computacionales. OWL Full permite una ontología para aumentar el significado del vocabulario preestablecido (TELLO, 2010 p. 5).

Debido a que es el más expresivo este lenguaje se usa en situaciones donde es más importante una extrema expresividad que la capacidad de razonamiento. Realiza unión de sintaxis de OWL y RDF (sin restricciones) (LUFERN, 2010 pág. 5) .

OWL es un lenguaje de Ontologías Web, los lenguajes anteriores han sido utilizados para desarrollar herramientas y ontologías destinadas a comunidades específicas (especialmente para ciencias y aplicaciones específicas de comercio electrónico), no fueron definidos para ser

compatibles con la arquitectura de la World Wide Web en general, y la Web Semántica en particular (W3C, 2004 pág. 1).

### **2.9.2 Capacidades de OWL**

OWL fue creado con la finalidad de representar el conocimiento y de esta manera inferir un nuevo conocimiento, para esto se emplean objetos y relaciones entre ellos. OWL se puede formular en RDF, por lo que se suele considerar una extensión de éste. Además incluye toda la capacidad expresiva de RDF y la extiende con la posibilidad de utilizar expresiones lógicas (W3C, 2004 pág. 1).

Permite la creación de clases y permite atribuir ciertas propiedades a las relaciones, como cardinalidad, simetría, transitividad o relaciones inversas. OWL rectifica esto proporcionando un lenguaje que utiliza la conexión proporcionada por RDF para añadir las siguientes capacidades a las ontologías (W3C, 2004 pág. 1):

- Capacidad de ser distribuidas a través de varios sistemas
- Escalable a las necesidades de la Web
- Compatible con los estándares Web de accesibilidad e internacionalización
- Abierto y extensible

### **2.9.3 Elementos de OWL**

Los principales elementos del lenguaje de ontologías para la web son los siguientes (PABLO DIEZ SUAREZ, 2008 pág. 6):

**Clases (Class):** Dentro de OWL una clase es un set de ceros o más instancias, y existe una clase universal Thing cuya extensión son todos los individuos de un modelo dado y las demás clases son subclases de Thing.

**Propiedades (Property):** Las propiedades son las que permiten las relaciones entre las clases de OWL, estas se representan por ObjectProperty y Dataproperty, y para distinguir el final se trabaja con designaciones como Dominio (domain) y rango (range).

Los DataProperty surgen de la asociación de la llamada ownedAttribute entre Class y Property y son traducidos como propiedades cuyo dominio es una clase y el rango es el tipo de la propiedad. Esto se puede entender mejor en el siguiente ejemplo, en donde se muestra el dominio y el rango de *ObjectProperty* TieneEsposa.

```
DataProperty( :name  
Domain ( :name :Persona)  
Range( :name xsd :String))
```

**Subclases/Subpropiedades (Subclass/Subproperty):** Dentro de OWL para representar una subclase se lo hace mediante subClassOf, como se observa en el siguiente ejemplo.

```
subClassOf( :Woman :Person)
```

**Relaciones entre clases (disjointWith/unionOf):** En OWL se utiliza la construcción disjointWith para declarar como clases disjuntas a las subclases. Mientras que para declarar que una superclase es la unión de sus subclases se utiliza unionOf.

```
disjointWith( :Mujer :Hombre)  
Class ( :Padres unionOf( :Madre :Padre))
```

**Cardinalidad:** En OWL se puede aplicar una restricción de cardinalidad en una relación, mediante un número máximo, mínimo o exacto dependiendo la necesidad, para ello se utilizan las restricciones minCardinality, maxCardinality y cardinality. A continuación se pueden observar tres ejemplos donde N representa el número máximo, mínimo o exacto de hijos que van a tener la relación Padres con tienenHijos.

```
minCardinality( N :tienenHijos :Padres)  
maxCardinality( N :tienenHijos :Padres)  
cardinality( N :tienenHijos :Padres)
```

## 2.10 RDF

RDF es un modelo estándar para el intercambio de datos en la Web. RDF facilita la fusión de datos incluso si los esquemas subyacentes son diferentes, y se apoya específicamente la evolución de los esquemas en el tiempo sin necesidad de ser cambiados todos los consumidores de datos (W3C, 2004). Por otra parte, RDF también lo podríamos usar para la construcción de ontologías, la diferencia con OWL radica en cuál es el nivel de expresión requerida por ejemplo, la cardinalidad en RDF sólo puede ser 1, mientras que en OWL no hay un límite establecido. Es importante tomar en cuenta, en la etapa de diseño, cuál es el nivel de expresión que se requiere y si vale la pena tomar el riesgo de generar ontologías que no garanticen ser computables (W3C, 2004 pág. 1).

RDF permite expresar las relaciones entre las cosas mediante la estandarización en un formato flexible, basada en el triple y luego proporcionar un vocabulario ("keywords" como RDF: type o rdfs: subclassOf) que se puede utilizar para decir cosas, es decir RDF tiene ciertas restricciones en cuanto al vocabulario utilizado. RDFS da por válido a todo lo que se pueda agregar casi cualquier tripleta que se desee (SEMANTICA, 2015 p. 2).

Por ejemplo, en RDF, se permite un free-for-all el mismo que puede ser una instancia de rdfs: Class. Es posible que decida decir que Beagle es un rdfs: Class y luego decir que Fido es una instancia de Beagle (SEMANTICA, 2015 p. 2):

*Ejemplo: Beagle RDF: rdfs Tipo: Clase*

*Ejemplo: Fido RDF: Tipo Ejemplo: Beagle*

### 2.10.1 Características de RDF

Como se mencionó anteriormente RDF es un modelo para el intercambio de datos, es por ello que se aplica en la web semántica, las principales características de este modelo son (MUÑOZ, 2007 pág. 3):

- Está particularmente indicado para representar Metadatos sobre recursos web, tales como títulos, autor, modificaciones de los datos de la página web, copyright y otras licencias de información sobre documentos web.
- Puede utilizarse para representar información sobre ciertas cosas que pueden ser identificadas en la Web, aunque no puedan ser directamente recuperadas en la misma.

- RDF está basado en la idea de que los objetos a describir poseen propiedades que a su vez tienen valores.
- Se trata de un modelo que permite definir relaciones semánticas entre distintas URIs asociándoles un conjunto de propiedades y valores.
- RDF es una evolución de WF (Warwick Framework), que permite que cada vocabulario de metadatos posea una sintaxis distinta.
- RDF mismo no es independiente de la sintaxis, ya que esta implementado sobre XML.
- Dado que RDF está implementado sobre XML, Los esquemas XML pueden utilizarse para validar las sintaxis de las expresiones RDF/XML. (TELLO, 2013 pág. 5)

### ***2.10.2 Formato de RDF***

Dentro de la semántica funcional existen tres aspectos del formato RDF los cuales son el modelo de datos, la sintaxis y el esquema.

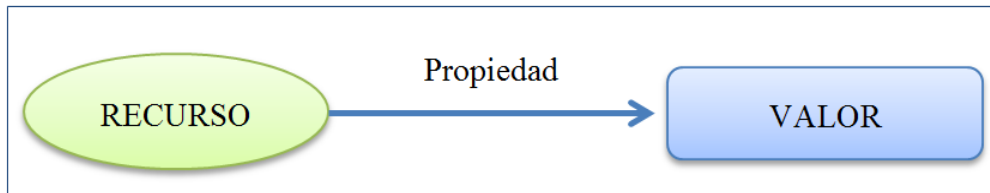
#### ***2.10.2.1 Modelo de datos***

En RDF existe un modelo formal para un recurso sus propiedades y también de sus valores. Por lo tanto se puede decir que RDF está construido en base a las siguientes reglas (MENDEZ, 2001 pág. 6):

- a. Un recurso es cualquier cosa que puede tener un URI, esto incluye todas las páginas web, todos los elementos individuales de cada documento XML y mucho más.  
El modelo de datos según la recomendación del W3C “RDF Especificación del Modelo y la Sintaxis” consiste en tres tipos de objetos.
- b. Una propiedad es un aspecto específico, característica, atributo, o relación utilizado para describir un recurso, por ejemplo autor o título. En muchos casos todo lo que nos importa en realidad es el nombre, pero una propiedad tiene un significado específico, define sus

valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades.

- c. Una sentencia consiste en la combinación de un recurso, una propiedad y un valor, es decir se trata de sujeto, predicado y objeto respectivamente, a estas tres partes se las llama un triple. Los triplos RDF pueden ser almacenados en bases de datos normales o en bases de datos especializadas.



**Figura 2-2.** Sentencia de RDF

Realizado por: Dalila Toledo

### **2.10.2.2 Sintaxis RDF**

Para la creación e intercambio de metadatos es necesario contar con una sintaxis definida en RDF, esta utiliza el lenguaje de marcado extensible XML. Existen dos tipos de sintaxis para codificar RDF, serializada y abreviada. La primera codificación permite expresar todas las capacidades sintácticas del modelo de una forma muy regular mientras que la segunda expresa de manera compacta un subconjunto del modelo de datos. Es posible que se trabajen por separado o mezclar ambas libremente a la hora de crear metadatos.

### **2.10.2.3 Esquema RDF**

Este esquema es una especie de diccionario donde se documentan las definiciones y las restricciones de uso de las propiedades. Este esquema proporciona mecanismos para la RECURSO VALOR Propiedad definición de clases, objetos y propiedades; y a su vez restricciones de dominio y rango sobre las propiedades (MUÑOZ, 2007 pág. 6).

#### 2.10.2.4 Contenedores en RDF

Muchas veces es necesario expresar que un recurso está relacionado con una colección de objetos, por ejemplo, para expresar que una métrica fue desarrollada por más de una persona, o para enumerar las distintas métricas que cuantifican un atributo de una entidad. Los contenedores RDF se usan para mantener tales listas de recursos o literales. RDF define tres tipos de objetos contenedores (GARCIA, 2008 pág. 4):

- **Bag:** Una lista desordenada de recursos o literales. Los bags (bolsa), se utilizan para indicar que una propiedad tiene múltiples valores y que no es significativo el orden en que se den tales valores.
- **Sequence:** Una lista ordenada de recursos o literales. Sequence se usa para manifestar que una propiedad tiene múltiples valores y que el orden de los valores es significativo.
- **Alternative:** Una lista de recursos o literales que representan alternativas para un valor (individual) de una propiedad. Una aplicación que utiliza una propiedad cuyo valor es una colección alternativa, sabe que se puede elegir como correcto cualquiera de los ítems en la lista.

### 2.11 SWRL

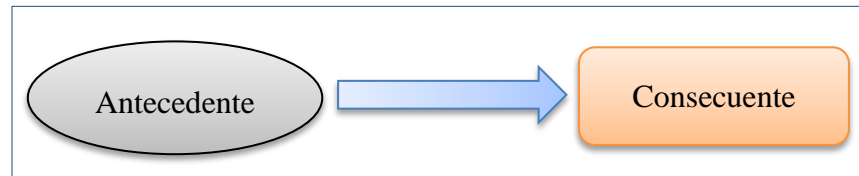
Este es un lenguaje que se extiende de OWL incluyendo Horn-like rules, que se pueden combinar con la base de conocimiento de OWL. SWRL ofrece una sintaxis abstracta de alto nivel que amplía la de OWL (LAGE, 2011 p. 6).

#### 2.11.1 Características de SWRL

Ya que se basa en OWL este es un lenguaje que también permite procesar la información requerida por las aplicaciones, pero a diferencia de OWL este lenguaje se basa en reglas específicas las cuales deben ser seguidas de acuerdo a las necesidades. Las principales características de este lenguaje son (ANTONIO GUERRERO, 2005 pág. 7):

- Mientras una ontología contiene una secuencia de hechos y axiomas de varias clases como de subclase, de equivalencia y de restricciones sobre propiedades, SWRL propone extender estos axiomas con axiomas de reglas.

- Un axioma de regla en SWRL consiste en un antecedente y en un consecuente, cada una de los cuales está compuesto por un conjunto de átomos.
- En la sintaxis perceptible por los seres humanos de SWRL, una regla tiene la siguiente forma:



**Figura 3-2.** Sintaxis de una regla en SWRL

Realizado por: Dalila Toledo

- El almacenamiento de información en la base de reglas se lo realiza con el propósito de verificar la integridad de la información y automatizar el control de los elementos gestionados.
- SWRL aporta sobre todo la definición de los átomos y su integración dentro de una ontología OWL.
- SWRL también permite definir explícitamente el comportamiento del gestor y de los objetos gestionados en el mismo lenguaje de ontologías.

### **2.11.2 Reglas de SWRL**

Las reglas dentro de SWRL son de gran importancia ya que nos permiten definir de mejor manera una sintaxis, dentro de SWRL existen reglas la cuales describen (LAGE, 2011 p. 7):

- Antecedente (body) Consecuente (head).
- Ambos formados por uno o más átomos.
- Un antecedente vacío implica verdad absoluta (true).
- Un consecuente vacío implica la falsedad del antecedente (false).



- Los átomos en estas reglas pueden ser de la forma:  $C(x)$ ,  $P(x,y)$ ,  $sameAs(x,y)$  o  $differentFrom(x,y)$ , donde  $C$  es una descripción.
- Se pueden utilizar referencias URI, que sirven para identificar una regla como por ejemplo:

$$\begin{aligned}
 rule & ::= '([ URIreference ] \{ annotation \} antecedent consequent )' \\
 antecedent & ::= 'Antecedent (\{ atom \})' \\
 consequent & ::= 'Consequent (\{ atom \})'
 \end{aligned}$$

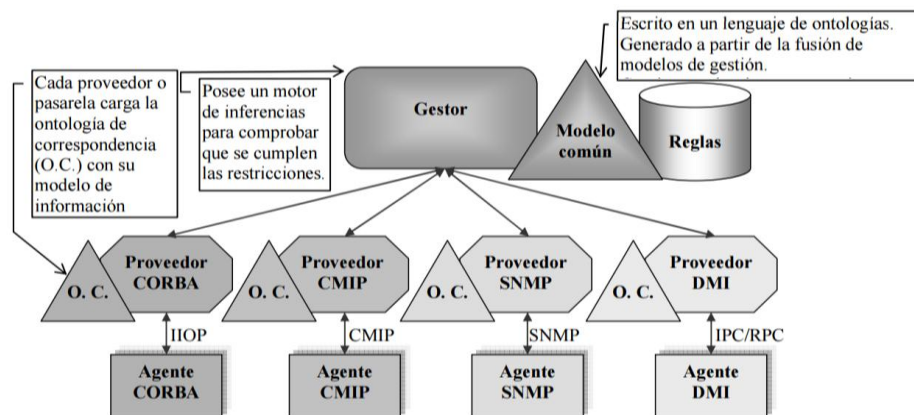
### 2.11.3 Especificación del comportamiento de gestión

Para poder especificar el comportamiento de gestión resulta conveniente clasificar que tipos pueden existir. En concreto, se han identificado tres tipos de comportamientos de gestión que son (ANTONIO GUERRERO, 2005 pág. 7):

1. Restricciones y reglas de comportamiento implícitas a los objetos modelados en las MIBs y Esquemas CIM.
2. Comportamiento explícito del gestor, según la arquitectura tradicional gestor-agente, en el que se indica cómo se comporta el gestor al obtener y analizar información de los agentes.
3. Políticas definidas explícitamente para especificar el comportamiento o configuración dinámica de los recursos gestionados.

En este marco de gestión semántica, las políticas se pueden definir en el mismo lenguaje de gestión, OWL+SWRL, que los objetos definidos en la base de información de gestión (MIB), con la ventaja que supone el trabajar con un modelo unificado.

Según la arquitectura de gestión propuesta, las definiciones de restricciones, reglas y políticas se almacenarían en la base de información de reglas, cuyo propósito es verificar la integridad de la información, y automatizar el control de los elementos gestionados (ANTONIO GUERRERO, 2005 pág. 7).



**Figura 4-2.** Arquitectura del gestor semántico de SWRL

Fuente: <http://newweb.dit.upm.es/~jlopez/publicaciones/jitel05-swrl.pdf>

## 2.12 LMS Moodle

Moodle es un software diseñado para ayudar a los educadores a crear cursos en línea de alta calidad y entornos de aprendizaje virtuales. Tales sistemas de aprendizaje en línea son también conocidos como VLEs o entornos virtuales de aprendizaje.

La palabra Moodle originalmente es un acrónimo de Modular Object-Oriented Dynamic Learning Environment que en español sería Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular (ANA TERESA GONZALEZ, 2009 pág. 6).

### 2.12.1 Características

Las características de LMS Moodle son las siguientes: (MEDIAWIKI, 2014 pág. 1)

- LMS Moodle está hecho en base a la pedagogía social constructivista, donde la comunicación tiene un espacio relevante en el camino de la construcción del conocimiento. Siendo el objetivo generar una experiencia de aprendizaje enriquecedora.
- La instalación requiere una plataforma que soporte PHP y la disponibilidad de una base de datos. Moodle tiene una capa de abstracción de bases de datos por lo que soporta los principales sistemas gestores de bases de datos.

- Se ha puesto énfasis en una seguridad sólida en toda la plataforma. Todos los formularios son revisados, las cookies cifradas, etc. La mayoría de las áreas de introducción de texto (materiales, mensajes de los foros, entradas de los diarios, etc.) pueden ser editadas usando el editor HTML, tan sencillo como cualquier editor de texto (MEDIAWIKI, 2014).

### **2.12.2 Ventajas de LMS Moodle**

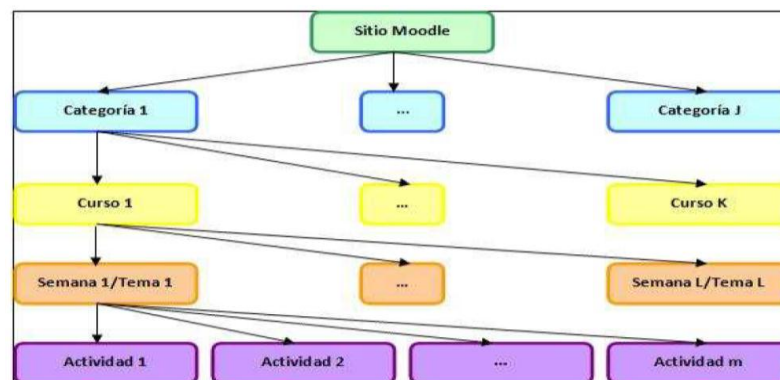
- Una de las características más atractivas de Moodle, que también aparece en otros gestores de contenido educativo, es la posibilidad de que los alumnos puedan participar en la creación de glosarios, y en todas las lecciones se generan automáticamente enlaces a las palabras incluidas en estos (HERNANDEZ, 2014 pág. 3).
- Las instituciones educativas podrán poner su Moodle local y así poder crear sus plataformas para cursos específicos y tan solo dando la dirección respecto a Moodle, se moverá en su mismo idioma y podrán abrirse los cursos a los alumnos que se encuentren en cualquier parte del planeta (HERNANDEZ, 2014 pág. 3).
- Software es libre.
- Soporta formato SCORM
- Incorpora numerosos módulos que son desarrollados y actualizados por una amplia comunidad.
- Dispone de una amplia gama de plantillas, recursos como chat, agenda y diseño de actividades.
- Permite distintos tipos de aprendizaje
- Es muy intuitivo. (HERNANDEZ, 2014 pág. 3)

### 2.12.3 Desventajas de LMS Moodle

- Algunas actividades pueden ser un poco mecánicas, dependiendo mucho del diseño instruccional. Por estar basado en tecnología PHP, la configuración de un servidor con muchos usuarios debe ser cuidadosa para obtener el mejor desempeño.
- Hay desventajas asociadas a la seguridad, dependiendo en dónde se esté alojando la instalación de Moodle y cuáles sean las políticas de seguridad y la infraestructura tecnológica con la cual se cuente durante la instalación. (MEDIAWIKI, 2014 pág. 1).
- La plataforma puede no ser relativamente fácil para muchos usuarios.
- Un fallo en los servidores o caída del servicio de internet, puede dejar al usuario inhabilitado para realizar sus actividades.
- En ocasiones el rol del administrador/profesor a veces es poco intuitivo y pesado de editar.
- Su instalación puede resultar bastante compleja. (HERNANDEZ, 2014 pág. 8).

### 2.12.4 Estructura de LMS Moodle

Un sitio LMS Moodle está diseñado para trabajar con una base de datos MySQL, la misma que puede albergar diversa información como categorías, cursos, temas (o semanas) y actividades. (ANA TERESA GONZALEZ, 2009 pág. 2)



**Figura 5-2.** Esquema de Moodle

Fuente: <http://www.aulafacil.com/uploads/cursos/2948/editor/147.es.jpg>

A continuación se procederá a explicar cada uno de los elementos que completan un sitio Moodle.  
(ANA TERESA GONZALEZ, 2009 pág. 1)

#### **2.12.4.1 Categorías**

Las categorías son los contenedores de información de más alto nivel, están formadas por cursos y sirven para organizarlos de manera que sean más fácilmente localizables por el alumno en la pantalla inicial de la aplicación.

#### **2.12.4.2 Cursos**

Los cursos son la parte más importante de la estructura de Moodle, son creados por los administradores del sitio y dirigidos por los usuarios que se establezcan como profesores del curso. Para que un alumno, dado de alta en el sistema, pueda acceder a un curso deberá estar matriculado en él.

Cuando un administrador crea un curso debe proceder a su configuración mediante un formulario proporcionado por Moodle en el que se establecen valores para distintos campos como, por ejemplo:

- Nombre y descripción.
- Formato del curso (semanal, por temas...)
- Número de semanas o temas.
- Fechas en las que permanecerá abierto el curso.

Una vez que un curso es creado, la matriculación la puede llevar a cabo cada usuario de forma independiente o bien el propio administrador, de forma masiva, sobre un grupo de usuarios del sistema, definiendo qué usuarios serán profesores y cuales alumnos. (ANA TERESA GONZALEZ, 2009 pág. 8)

#### **2.12.4.3 Semanas y Temas**

La organización de un curso podrá llevarse a cabo por semanas o bien por temas, según la preferencia del profesorado. Tras la creación del curso, su configuración, y la matriculación de

usuarios, estos podrán acceder al mismo y observar una serie de bloques diferenciados que representan las semanas del curso o temas, según el formato que se haya establecido. Cada uno de estos bloques contendrá, a partir del momento en el que el profesor las añade, distintos tipos de actividades que los alumnos deberán realizar para su evaluación (ANA TERESA GONZALEZ, 2009 pág. 9).

## CAPÍTULO III

### 3. ANÁLISIS DE TECNOLOGÍAS RDF, OWL Y SWRL

#### 3.1 Introducción

Para realizar el análisis de las diferentes tecnologías se requiere definir los índices e indicadores, los mismos que ayudaran a la mejor determinacion de los resultados.

#### 3.2 Determinación de índices y parámetros de análisis

Para realizar el estudio de comparación entre las tecnologías de web semántica OWL, RDF y SWRL se establecieron varios parámetros, los cuales permitieron determinar cuál de las tecnologías antes mencionadas es la idónea para la aplicación de la web semántica en la creación de ontologías de cursos virtuales, a partir de una base de datos LMS.

Los parámetros de comparación definidos son los siguientes:

*Tabla 1-3. Índices y parámetros de comparación*

Índice	Parámetros
1. Productividad	<ul style="list-style-type: none"><li>• Tiempo de procesamiento</li><li>• Consumo de Recursos</li></ul>
2. Usabilidad	<ul style="list-style-type: none"><li>• Facilidad de aprendizaje</li><li>• Interoperabilidad</li></ul>
3. Procesamiento	<ul style="list-style-type: none"><li>• Acceso a Datos</li><li>• Asociación de información</li></ul>
4. Programación	<ul style="list-style-type: none"><li>• Familiaridad con otros lenguajes</li><li>• Tiempo de desarrollo</li></ul>

Realizado por: Dalila Toledo

### 3.2.1 Descripción de indicadores y parámetros

A continuación se realiza una breve descripción de cada uno de los indicadores y sus respectivos parámetros.

#### **INDICADOR 1: PRODUCTIVIDAD**

Se refiere principalmente a las facilidades que una herramienta determinada presta ante el desarrollo de un nuevo producto, el mismo que sea capaz de cumplir con las necesidades del usuario en un determinado tiempo y a su vez aproveche los recursos existentes disponibles durante el ciclo de desarrollo y en su aplicación práctica.

- **Tiempo de Procesamiento:** Un aspecto muy importante a tomar en cuenta dentro de la productividad es el tiempo de procesamiento ya que en términos de productividad mientras mejor sea aprovechado el tiempo de procesamiento más productivo será el sistema y las herramientas utilizadas para su creación, es decir mientras mejor sea su conexión con la base de datos mejor será su tiempo de respuesta. Por lo tanto mediante este parámetro se puede determinar la tecnología que es la más productiva en cuanto a tiempo se refiere.
- **Consumo de Recursos:** Este parámetro permite valorar el uso de los recursos informáticos disponibles, permitiendo así determinar cuál es el más óptimo de acuerdo a las necesidades del usuario, en este caso el consumo de procesador y memoria para la obtención de información de la base es lo principal.

#### **INDICADOR 2: USABILIDAD**

Este indicador trata a cerca de la rapidez y facilidad con que las personas llevan a cabo sus tareas propias a través del uso del producto. Es un intento de cuantificar “lo amigable que puede ser un sistema con el usuario”.

- **Facilidad de Aprendizaje:** Una de las características principales al momento de desarrollar un sistema es la facilidad de aprendizaje que este brinda al usuario, por lo tanto este parámetro permite definir la tecnología más amigable con el usuario al momento de la extracción de información desde la base de datos.
- **Interoperabilidad:** Se refiere a la capacidad que tiene un sistema o programa para poder ejecutarse y poder operar independientemente, cumpliendo todas las funciones y así tener acceso completo a la información disponible. Característica dada tanto por la plataforma



como por el software que trabaja u opera en esta. Es el esfuerzo requerido para acoplar un sistema con otro u otros.

### **INDICADOR 3: PROCESAMIENTO**

Mediante este indicador se mide la calidad de procesamiento y asociación de la información disponible, que tiene cada una de las tecnologías.

- **Acceso a Datos:** Este parámetro es una parte fundamental dentro de las aplicaciones informáticas, porque permite determinar la tecnología idónea al momento de acceder y manipular la información almacenada en bases de datos o en archivos XML, a través de la utilización de consultas.
- **Asociación de información:** Una característica importante al trabajar con la web semántica, es la capacidad en la asociación de información que tiene cada tecnología, ya que para la creación de las ontologías se requiere una gran capacidad de asociación de los datos almacenados.

### **INDICADOR 4: PROGRAMACIÓN**

Este indicador nos permite valorar de manera técnica al proceso por el cual se crea, prueba, depura y mantiene el código fuente de un programa informático. Es decir la programación es un aspecto básico dentro de la informática.

- **Familiaridad con otros lenguajes:** La calidad de las aplicaciones desarrolladas depende principalmente de los lenguajes de programación utilizados y su compatibilidad con otros lenguajes, ya que hasta los procesos más complejos en la obtención de información pueden ser tratados mediante técnicas de programación.
- **Tiempo de desarrollo:** El tiempo empleado para el desarrollo de un sistema pesa mucho en cuanto a su valoración, ya que mientras mejor sea aprovechado el tiempo más óptima será la herramienta de desarrollo utilizada.

#### **3.2.2 *Métodos para la evaluación de resultados***

Las tres tecnologías de la web semántica fueron evaluadas mediante una comparativa entre los diferentes parámetros anteriormente establecidos. Cuyos resultados se irán registrando en tablas durante el proceso de estudio.

Por cada parámetro de comparación tecnología obtendrá una calificación la misma que la final se resumirá en un resultado global el cual nos permitirá decidir cuál es la tecnología más idónea para la aplicación en la web semántica.

### 3.2.3 Valoración de Parámetros

Para realizar el presente estudio se tomó como referencia los siguientes valores tanto cualitativos como cuantitativos, para de esta manera tener una mejor percepción de los resultados. Los valores han sido establecidos de la siguiente manera:

*Tabla 2-3. Valoración de parámetros*

Valor Cualitativo	Valor Cuantitativo
No dispone	0
Regular	1
Bueno	2
Muy Bueno	3
Excelente	4

Realizado por: Dalila Toledo

Los valores van desde 0 a 4, donde 0 es el valor más bajo y que representa la ausencia de un índice de comparación, 1 representa que existe muy poca presencia del índice de valoración, y 4 representa que existe el índice de valoración y que satisface completamente a las expectativas.

Una vez obtenidos los respectivos valores de cada uno de los indicadores, se procede a la obtención de resultados tanto cualitativos como cuantitativos mediante la sumatoria de dichos valores, para lo cual se trabaja con la siguiente formula:

$$ValorTotal = \sum_{i=1}^n VP_i$$

Dónde:

*i*: Representa el parámetro específico.

*n*: Representa el total de parámetros evaluados.

*VP*: Representa el valor obtenido por cada parámetro.

*ValorTotal*: Representa el valor total o resultado al cual deseamos llegar.

### 3.3 Análisis de los índices y parámetros de comparación

El análisis comparativo de los parámetros previamente establecidos se lo realizó en base a la teoría obtenida durante la fase de investigación bibliográfica así como también de lo observado en el proceso de desarrollo y pruebas de los prototipos construidos; y se los valorizo en base a criterios de evaluación que se presentan en forma de tablas con rangos de valores numéricos que están relacionados con valores cualitativos.

#### 3.3.1 Indicador 1: Productividad

Para medir la productividad se tomó en cuenta los parámetros:

- Tiempo de Procesamiento
- Consumo de Recursos

##### 3.3.1.1 Parámetro Tiempo de Procesamiento

Para la medición tiempo de procesamiento se definió los siguientes criterios de evaluación:

*Tabla 3-3. Criterios de evaluación del parámetro de Tiempo de Procesamiento*

Valor Cuantitativo	Valor Cualitativo
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente

Realizado por: Dalila Toledo

Para la medición del tiempo de procesamiento se tomó en cuenta, el tiempo de acceso a la base, tiempo de asociación de información. El análisis se lo realizo de la siguiente manera como se indica en la Tabla 5-3. El porcentaje de interoperabilidad se lo representa según la cantidad de (\*), teniendo en cuenta que los valores pueden ser Bueno, Regular y malo, de la siguiente manera:

**Tabla 4-3. Simbolización de Tiempo de Procesamiento**

Simbolización	Equivalencia	Valores referencia
***	Bueno	0 seg – 5 seg
**	Regular	6 seg – 10 seg
*	Malo	+10 seg

Realizado por: Dalila Toledo

**Tabla 5-3. Datos obtenidos del Parámetro Tiempo de Procesamiento**

	Tiempo de Procesamiento		Total Tiempo de Procesamiento	Valor Cuantitativo	Valor Cualitativo
	Tiempo de acceso a la base	Tiempo de asociación de información			
<b>OWL</b>	***	**	5	4	Excelente
<b>RDF</b>	**	**	4	3	Muy Bueno
<b>SWRL</b>	**	**	4	3	Muy Bueno

Realizado por: Dalila Toledo

## Interpretación

El análisis del tiempo de procesamiento se fundamentó en el criterio de mientras menor sea el tiempo de respuesta, más alta es su valoración, y mientras menor sea el tiempo de respuesta menor es su valoración.

Según los valores obtenidos se puede observar que la tecnología OWL ha obtenido una valoración cualitativa de EXCELENTE con un valor cuantitativo de 4, tomando en cuenta que de acuerdo a su estructura el tiempo de acceso a la base de datos es mucho más eficaz que las otras tecnologías estudiadas. Mientras que las tecnologías RDF y SWRL obtuvieron una valoración cualitativa de MUY BUENO con un valor cuantitativo de 3, ya que sus tiempos de respuesta son muy aceptables.

### 3.3.1.2 *Parámetro Consumo de Recursos*

Para la medición del consumo de recursos se definieron los siguientes criterios de evaluación:

*Tabla 6-3. Criterios de evaluación del parámetro de Consumo de Recursos*

Valor Cuantitativo	Valor Cualitativo
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente

**Realizado por:** Dalila Toledo

Para la medición del Consumo de recursos se tomó en cuenta el consumo de memoria RAM, utilización de disco duro, y porcentaje de uso del CPU. El análisis se lo realizo de la siguiente manera como se indica en la Tabla 8-3. El porcentaje de consumo de cada recurso se lo representa según la cantidad de (\*), teniendo en cuenta que el consumo puede ser Alto, Medio, Aceptable y Bajo, de la siguiente manera:

*Tabla 7-3. Simbolización de Consumo de Recursos*

Simbolización	Equivalencia	Porcentajes de Referencia
****	Alto	66%+
***	Medio	46% - 65%
**	Aceptable	21% - 45%
*	Bajo	0% - 20%

**Realizado por:** Dalila Toledo

**Tabla 8-3. Datos obtenidos del Parámetro Consumo de Recursos**

	Consumo de Recursos			Total Consumo de Recursos	Valor Cuantitativo	Valor Cualitativo
	Consumo de memoria RAM	Espacio en el disco duro	Uso del CPU			
<b>OWL</b>	**	*	**	5	3	Muy Bueno
<b>RDF</b>	**	**	**	6	2	Bueno
<b>SWRL</b>	**	**	***	7	1	Regular

**Realizado por:** Dalila Toledo

### **Interpretación**

El análisis se fundamentó en el criterio de mientras menos recursos utilice la tecnología estudiada, mejor será el rendimiento y por lo tanto mientras más recursos requiera menor será el rendimiento.

Según los valores obtenidos se puede observar que la tecnología OWL ha obtenido una valoración cualitativa de MUY BUENO con un valor cuantitativo de 3, puesto que es la tecnología que presenta un mejor porcentaje de consumo en cuanto al Espacio en el disco duro. Mientras que la tecnología RDF obtuvo una valoración cualitativa de 2 cuyo equivalente cualitativo es de BUENO, y la tecnología SWRL obtuvo una valoración cualitativa de REGULAR con un valor cuantitativo de 1, ya que utilizan un mayor porcentaje de recursos. RDF y SWRL requieren de más espacio en el disco duro, y SWRL también requiere mayor uso de CPU para funcionar correctamente.

#### **3.3.1.3 Resultados del Indicador de Productividad**

Una vez analizados los parámetros de tiempo de procesamiento y consumo de recursos, se procedió a la obtención del resultado en cuanto al índice de Productividad, teniendo como consecuencia lo siguiente:

Para la evaluación del índice de Productividad se tomaron como base los siguientes criterios:

**Tabla 9-3. Criterios de Evaluación de la Productividad**

Valor Cuantitativo	Valor Cualitativo	Porcentaje (%)
1	Regular	0% - 25%
2	Bueno	26% - 50%
3	Muy Bueno	51% - 75%
4	Excelente	76% - 100%

**Realizado por:** Dalila Toledo

Para determinar el valor cuantitativo de los datos obtenidos se aplica una regla de tres, con los siguientes datos:

El 100% son 8 puntos ya que es el valor máximo que resulta de la suma de los dos parámetros.

A es el total resultante de cada tecnología.

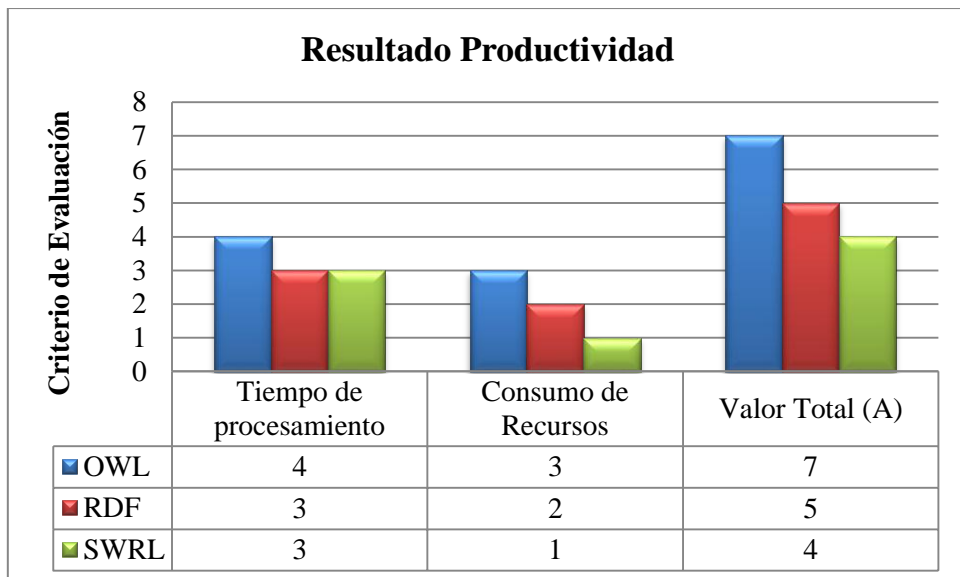
Entonces la Fórmula Utilizada es:

$$\text{Valor cuantitativo} = \frac{Ax100}{8}$$

**Tabla 10-3. Resultados obtenidos del Índice Productividad**

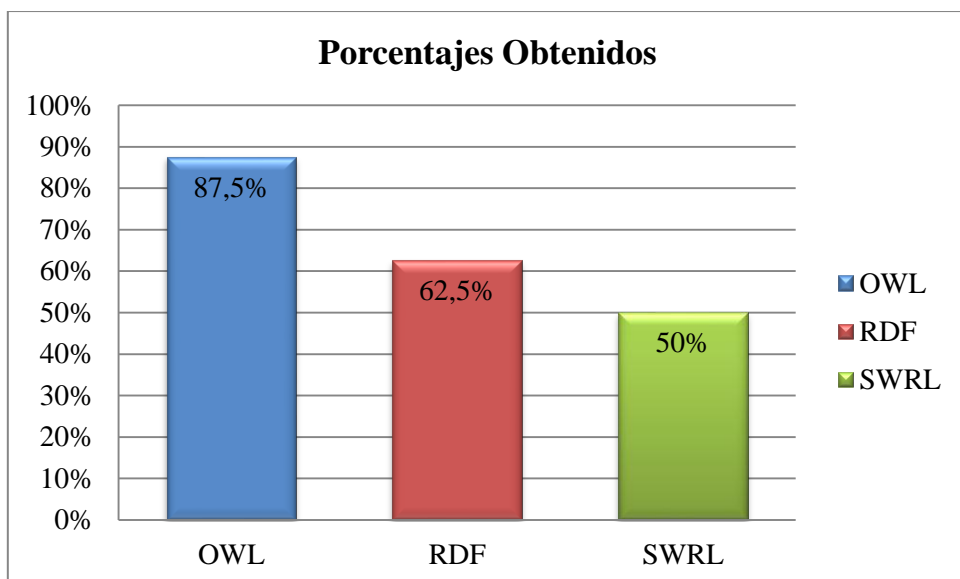
Tecnologías Parámetros	OWL	RDF	SWRL
Tiempo de procesamiento	4	3	3
Consumo de Recursos	3	2	1
<b>Valor Total (A)</b>	<b>7</b>	<b>5</b>	<b>4</b>
<b>Valor Cuantitativo</b>	<b>87.5%</b>	<b>62.5%</b>	<b>50%</b>
<b>Valor Cualitativo</b>	<b>Excelente</b>	<b>Muy Bueno</b>	<b>Bueno</b>

**Realizado por:** Dalila Toledo



**Figura 1-3.** Resultado del Índice Productividad

Realizado por: Dalila Toledo



**Figura 2-3.** Porcentajes obtenidos de índice Productividad

Realizado por: Dalila Toledo



## Interpretación

Para el análisis de este indicador se tomó en cuenta el tiempo de procesamiento y el consumo de recursos de las tecnologías. El Consumo de recursos se valoró tomando en cuenta que cuando menor es el consumo, mayor es la puntuación. Mientras que en lo referente al Tiempo de Procesamiento se valoró de la siguiente manera, mientras menor es el tiempo requerido más alta es su puntuación.

Por lo tanto los resultados obtenidos muestran que la tecnología que nos brinda una mayor productividad es OWL con una puntuación cualitativa de EXCELENTE y cuyo valor cuantitativo es del 87.5%, mientras que las tecnologías menos productivas son RDF y SWRL las mismas que obtuvieron una puntuación cualitativa de MUY BUENO y BUENO respectivamente, y cuyos valores cuantitativos son del 62.5% para RDF y el 50% para SWRL.

### 3.3.2 *Indicador 2: Usabilidad*

Para medir la usabilidad se tomó en cuenta los parámetros:

- Facilidad de aprendizaje
- Interoperabilidad

#### 3.3.2.1 *Parámetro Facilidad de Aprendizaje*

Para realizar la medición en cuanto a la facilidad de aprendizaje se definieron los siguientes criterios de evaluación:

*Tabla 11-3. Criterios de evaluación para la Facilidad de Aprendizaje*

Valor Cuantitativo	Valor Cualitativo
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente

Realizado por: Dalila Toledo

La medición de la Facilidad de aprendizaje se basó en tres factores importantes desde el punto de vista de programación, los cuales son: el lenguaje de programación utilizado, arquitectura utilizada y estructuración de archivos. El análisis se lo realizó de la siguiente manera como se indica en la Tabla 13-3. El porcentaje de facilidad de aprendizaje se lo representa según la cantidad de (\*), teniendo en cuenta que los valores pueden ser Fácil, Medio y Difícil, de la siguiente manera:

**Tabla 12-3. Simbolización de Facilidad de Aprendizaje**

Simbolización	Equivalencia	Días Aprendizaje
*	Fácil	1 – 2
**	Medio	3 – 4
***	Difícil	5+

**Realizado por:** Dalila Toledo

**Tabla 13-3. Datos obtenidos del Parámetro Facilidad de Aprendizaje**

	Facilidad de Aprendizaje			Total Facilidad de Aprendizaje	Valor Cuantitativo	Valor Cualitativo
	Lenguaje de programación utilizado	Arquitectura utilizada	Estructura de Archivos			
<b>OWL</b>	**	**	*	5	3	Muy Bueno
<b>RDF</b>	**	*	*	4	4	Excelente
<b>SWRL</b>	**	**	**	6	2	Bueno

**Realizado por:** Dalila Toledo

## **Interpretación**

El análisis se fundamentó tomando el criterio de mientras más familiar sea el lenguaje, arquitectura y estructura utilizadas por la tecnología estudiada mayor será la valoración y por ende mientras menos familiar sea, menor será su valoración. Según los valores obtenidos se puede observar que la tecnología RDF ha obtenido una valoración cualitativa de EXCELENTE, con un valor cuantitativo de 4, puesto que es la tecnología cuya arquitectura y estructura de archivos utilizada es más familiar. Mientras que las tecnologías OWL y SWRL obtuvieron una valoración cualitativa de MUY BUENO y BUENO respectivamente. Donde OWL obtuvo un valor cuantitativo de 3, ya que

la estructura de los archivos es similar a la estructura de RDF, mientras que SWRL obtuvo un valor cuantitativo de 2 ya que su estructura de archivos y arquitectura utilizada es más compleja que las otras tecnologías estudiadas.

### 3.3.2.2 *Parámetro Interoperabilidad*

Para la medición la interoperabilidad se definió los siguientes criterios de evaluación:

**Tabla 14-3. Criterios de evaluación del parámetro de Interoperabilidad**

Valor Cuantitativo	Valor Cualitativo
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente

**Realizado por:** Dalila Toledo

Para la medición de la interoperabilidad se tomó en cuenta el Manejo de componentes, la Gestión de Ingeniería y Revisión Automática de Sintaxis. El análisis se lo realizó de la siguiente manera como se indica en la Tabla 16-3. El porcentaje de interoperabilidad se lo representa teniendo en cuenta que los valores pueden ser Posee y No Posee, de la siguiente manera:

**Tabla 15-3. Simbolización de Interoperabilidad**

Simbolización	Equivalencia
✓	Posee
--	No Posee

**Realizado por:** Dalila Toledo

**Tabla 16-3. Datos obtenidos del Parámetro Interoperabilidad**

	Interoperabilidad			Total Interoperabilidad	Valor Cuantitativo	Valor Cualitativo
	Manejo de Componentes	Gestión de Ingeniería de Software	Revisión Automática de Sintaxis			
<b>OWL</b>	✓	✓	✓	3	4	Excelente
<b>RDF</b>	✓	✓	✓	3	4	Excelente
<b>SWRL</b>	✓	✓	--	2	3	Muy Bueno

Realizado por: Dalila Toledo

### **Interpretación**

El análisis de la interoperabilidad se fundamentó en el criterio de si posee o no posee una característica como la gestión de ingeniería de software.

Según los valores obtenidos se puede observar que las tecnologías estudiadas OWL y RDF obtuvieron una valoración cualitativa de EXCELENTE con un valor cuantitativo de 4, puesto que ambas tecnologías brindan la opción de Revisión de sintaxis. Mientras que la tecnología SWRL obtuvo una valoración cualitativa de MUY BUENO con un valor cuantitativo de 3, ya que no brinda la opción de revisión de sintaxis como si poseen las otras tecnologías estudiadas.

#### **3.3.2.3 Resultados del Indicador de Usabilidad**

Una vez analizados los parámetros de facilidad de aprendizaje e Interoperabilidad, se procedió a la obtención del resultado en cuanto al índice de Usabilidad, teniendo como consecuencia lo siguiente:

**Tabla 17-3. Criterios de Evaluación de la usabilidad**

Valor Cuantitativo	Valor Cualitativo	Porcentaje (%)
1	Regular	0% - 25%
2	Bueno	26% - 50%
3	Muy Bueno	51% - 75%
4	Excelente	76% - 100%

Realizado por: Dalila Toledo

Para determinar el valor cuantitativo de los datos obtenidos se aplica una regla de tres, con los siguientes datos:

El 100% son 8 puntos ya que es el valor máximo que resulta de la suma de los dos parámetros.

**A** es el total resultante de cada tecnología.

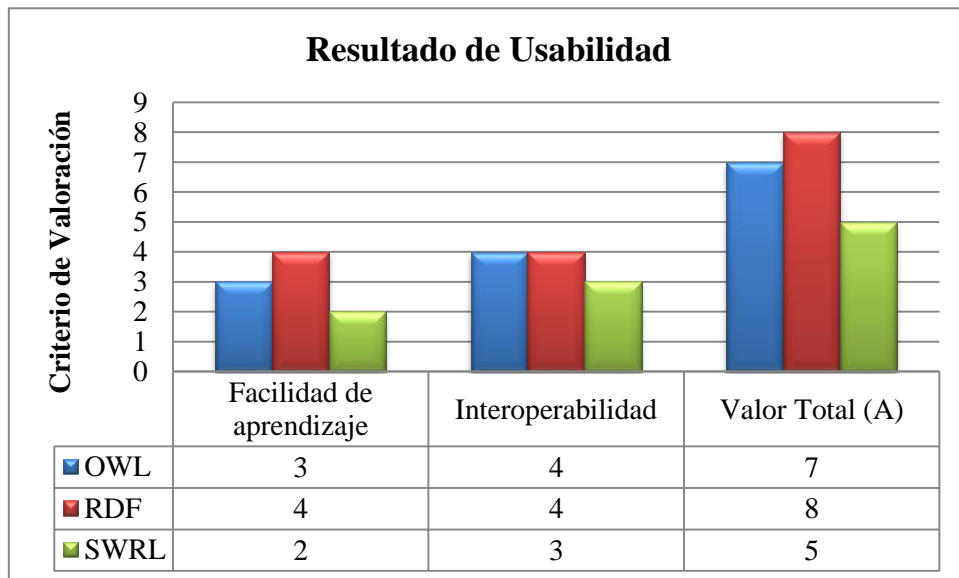
Entonces la Fórmula Utilizada es:

$$\text{Valor cuantitativo} = \frac{Ax100}{8}$$

**Tabla 18-3. Resultados obtenidos del Índice Usabilidad**

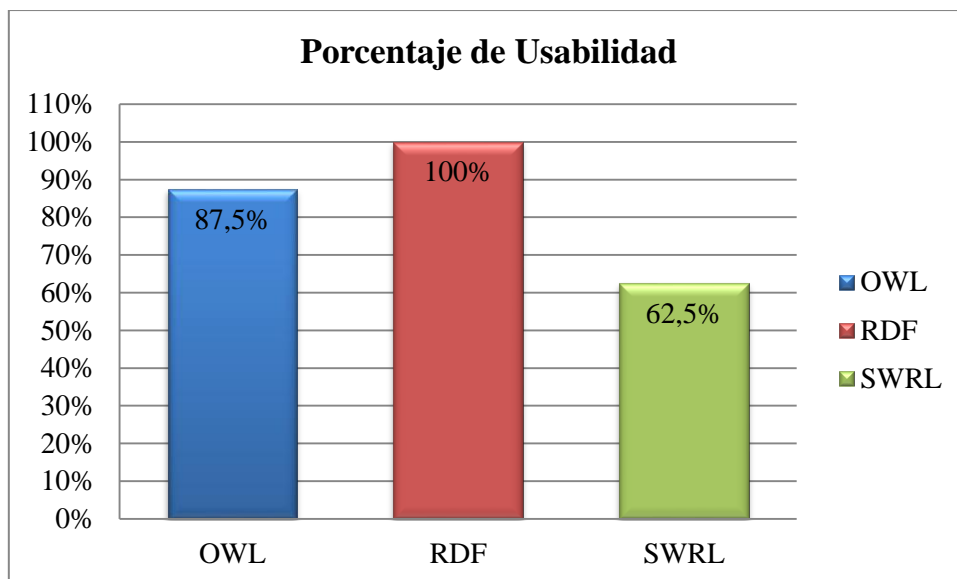
Tecnologías \ Parámetros	OWL	RDF	SWRL
Facilidad de aprendizaje	3	4	2
Interoperabilidad	4	4	3
<b>Valor Total (A)</b>	<b>7</b>	<b>8</b>	<b>5</b>
<b>Valor Cuantitativo</b>	<b>87.5%</b>	<b>100%</b>	<b>62.5%</b>
<b>Valor Cualitativo</b>	<b>Excelente</b>	<b>Excelente</b>	<b>Muy bueno</b>

Realizado por: Dalila Toledo



**Figura 3-3. Resultado del Índice Usabilidad**

Realizado por: Dalila Toledo



**Figura 4-3.** Porcentajes obtenidos del índice Usabilidad

Realizado por: Dalila Toledo

### Interpretación

Para el análisis de la usabilidad se tomó en cuenta la Facilidad de aprendizaje y la interoperabilidad de las tecnologías. La facilidad de aprendizaje se valoró tomando en cuenta que cuando más fácil es el aprendizaje, mayor es la puntuación. Mientras que en lo referente a la interoperabilidad se valoró con el criterio de si posee o no posee cierta característica.

Por lo tanto los resultados obtenidos muestran que la tecnología que nos brinda una mayor usabilidad es RDF con una puntuación cualitativa de EXCELENTE y cuyo valor cuantitativo es del 100%, mientras que la tecnología OWL también obtuvo la puntuación cualitativa de EXCELENTE pero con un porcentaje cuantitativo del 87.5% debido a que la facilidad de aprendizaje es menor que en la tecnología RDF. Por otro lado la tecnología SWRL obtuvo la puntuación más baja con un valor cualitativo de MUY BUENO y su valor cuantitativo de 62.5%.

#### 3.3.3 Indicador 3: Procesamiento

Para medir la capacidad de procesamiento se tomó en cuenta los parámetros:

- Acceso a Datos
- Asociación de información

### 3.3.3.1 *Parámetro Acceso a datos*

Para realizar la medición en cuanto acceso a datos se definieron los siguientes criterios de evaluación:

*Tabla 19-3. Criterios de evaluación parámetro Acceso a Datos*

<b>Valor Cuantitativo</b>	<b>Valor Cualitativo</b>
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente

**Realizado por:** Dalila Toledo

La medición acceso a datos se basó en el Vocabulario utilizado, la Rigidez y Las Anotaciones de data y meta data. El análisis se lo realizo de la siguiente manera como se indica en la Tabla 21-3. El porcentaje de acceso a datos se lo representa según la cantidad de (\*), teniendo en cuenta que los valores pueden ser Poco, Medio y Mucho, de la siguiente manera:

*Tabla 20-3. Simbolización de Acceso a Datos*

<b>Simbolización</b>	<b>Equivalencia</b>
*	Poco
**	Medio
***	Mucho

**Realizado por:** Dalila Toledo

**Tabla 21-3. Datos obtenidos del Parámetro Acceso a Datos**

	Acceso a Datos			Total Acceso a Datos	Valor Cuantitativo	Valor Cualitativo
	Vocabulario	Rigidez	Anotaciones de Data y Meta data			
<b>OWL</b>	***	**	***	8	4	Excelente
<b>RDF</b>	**	***	*	6	3	Muy Bueno
<b>SWRL</b>	**	**	*	5	2	Bueno

**Realizado por:** Dalila Toledo

### **Interpretación**

El análisis se fundamentó tomando el criterio de si posee o no una acción específica en cuanto al acceso y manejo de información.

Según los valores obtenidos se puede observar que OWL le da un vocabulario mucho más grande para jugar, lo que hace que sea fácil de decir cualquier cosa que quieras decir sobre su modelo de datos. Además, OWL le permite expresar fácilmente las relaciones entre las diferentes ontologías usando un marco de anotación estándar. Por lo tanto la mejor tecnología en cuanto a acceso a datos es OWL la misma que obtuvo una calificación cualitativa de EXCELENTE cuyo equivalente cuantitativo fue de 4. Mientras tanto las tecnologías de RDF y SWRL obtuvieron una calificación cualitativa de MUY BUENO y BUENO respectivamente.

#### **3.3.3.2 Parámetro Asociación de Información**

Para la evaluación de la asociación de información se definió los siguientes criterios de evaluación:

**Tabla 22-3. Criterios de evaluación del parámetro de Asociación de Información**

Valor Cuantitativo	Valor Cualitativo
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente



Realizado por: Dalila Toledo

Para la medición de la Asociación de Información se tomó en cuenta relación entre tablas, el uso de sentencias y el uso de formatos de reglas. El análisis se lo realizó de la siguiente manera como se indica en la Tabla 24-3. El porcentaje de la Asociación de Información se lo representa teniendo en cuenta que los valores pueden ser Posee y No posee, de la siguiente manera:

*Tabla 23-3. Simbolización de Asociación de Información*

Simbolización	Equivalencia
✓	Posee
--	No posee

Realizado por: Dalila Toledo

*Tabla 24-3. Datos obtenidos del Parámetro Asociación de Información*

	Asociación de información			Total Asociación de información	Valor Cuantitativo	Valor Cualitativo
	Relación entre tablas	Uso de sentencias	Uso de formatos de reglas			
<b>OWL</b>	--	✓	✓	2	2	Excelente
<b>RDF</b>	✓	✓	✓	3	4	Excelente
<b>SWRL</b>	✓	✓	--	2	2	Bueno

Realizado por: Dalila Toledo

## Interpretación

El análisis de la asociación de información se fundamentó en el criterio de si posee o no una característica como la relación de tablas, el uso de sentencias y uso de formatos de reglas.

Según los valores obtenidos se puede observar que tanto las tecnologías OWL y RDF obtuvieron una valoración cualitativa de EXCELENTE con un valor cuantitativo de 4, puesto que ambas tecnologías poseen las características especificadas. Mientras que la tecnología SWRL obtuvo una valoración cualitativa de BUENO con un valor cuantitativo de 2, ya que no posee la característica del uso de formatos de reglas así como las otras tecnologías estudiadas.

### 3.3.3.3 Resultados del Indicador de Procesamiento

Una vez analizados los parámetros de Acceso a datos, Asociación de información y Tiempo de procesamiento, se procedió a la obtención del resultado en cuanto al índice de Procesamiento, teniendo como consecuencia lo siguiente:

**Tabla 25-3. Criterios de Evaluación del Procesamiento**

Valor Cuantitativo	Valor Cualitativo	Porcentaje (%)
1	Regular	0% - 25%
2	Bueno	26% - 50%
3	Muy Bueno	51% - 75%
4	Excelente	76% - 100%

**Realizado por:** Dalila Toledo

Para determinar el valor cuantitativo de los datos obtenidos se aplica una regla de tres, con los siguientes datos:

El 100% son 8 puntos ya que es el valor máximo que resulta de la suma de los tres parámetros.

A es el total resultante de cada tecnología.

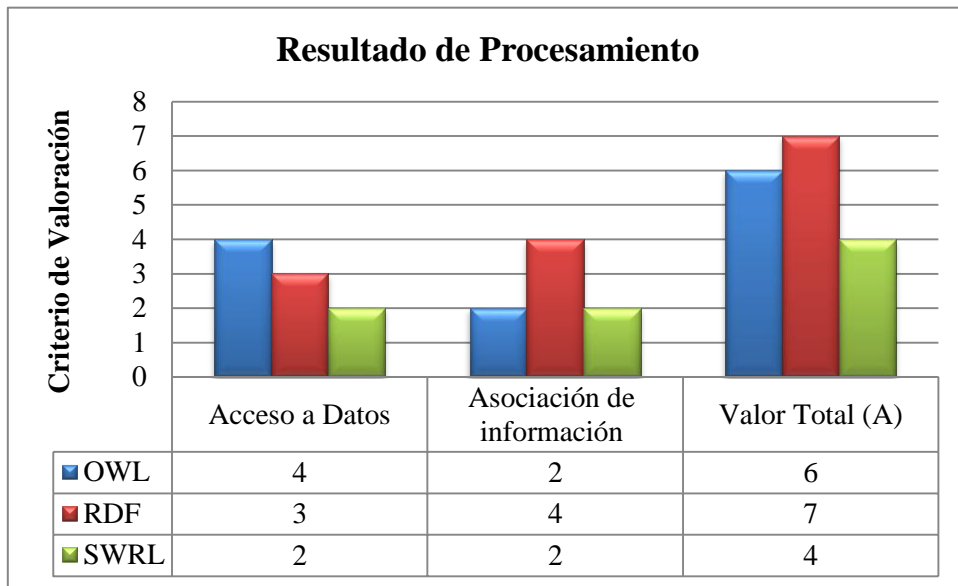
Entonces la Fórmula Utilizada es:

$$\text{Valor cuantitativo} = \frac{Ax100}{8}$$

**Tabla 26-3. Resultados obtenidos del Índice Procesamiento**

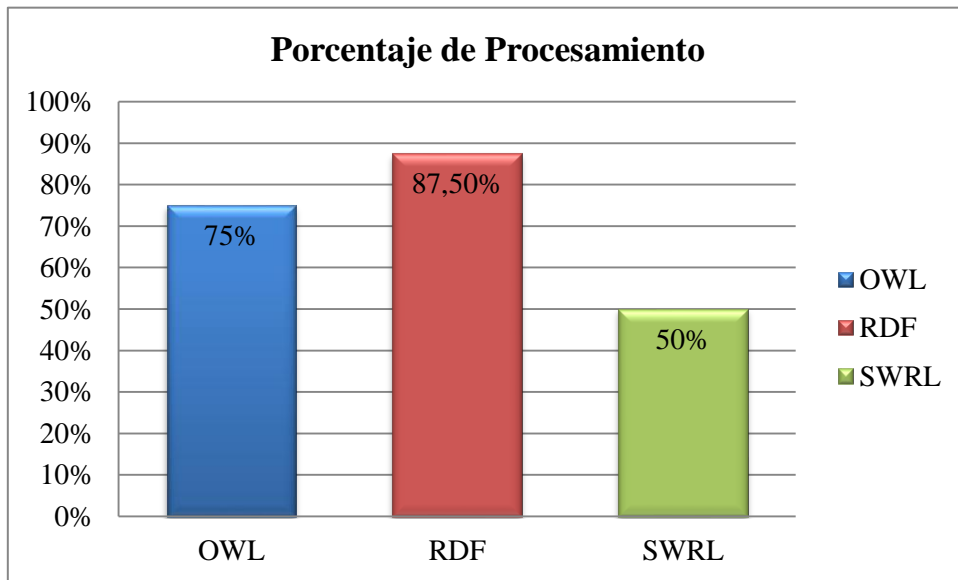
<b>Tecnologías</b> <b>Parámetros</b>	<b>OWL</b>	<b>RDF</b>	<b>SWRL</b>
Acceso a Datos	4	3	2
Asociación de información	2	4	2
<b>Valor Total (A)</b>	<b>6</b>	<b>7</b>	<b>4</b>
<b>Valor Cuantitativo</b>	<b>75%</b>	<b>87.5%</b>	<b>50%</b>
<b>Valor Cualitativo</b>	<b>Muy Bueno</b>	<b>Excelente</b>	<b>Bueno</b>

**Realizado por:** Dalila Toledo



**Figura 5-3.** Resultado del Índice Procesamiento

Realizado por: Dalila Toledo



**Figura 6-3.** Porcentajes obtenidos del Índice Procesamiento

Realizado por: Dalila Toledo

## Interpretación

Para el análisis de este indicador se tomó en cuenta el Acceso a datos, la asociación de información y el tiempo de procesamiento de las tecnologías. El acceso a datos se valoró tomando en cuenta que cuando mayor es la capacidad de acceso a los datos de una base, mayor es la puntuación. La asociación de información se valoró tomando en cuenta los métodos de asociación de información, es decir mientras más fácil sea la asociación mayor es el puntaje. En lo referente al tiempo de procesamiento se valoró de la siguiente manera, mientras menor es el tiempo de procesamiento más alta es su puntuación.

Por lo tanto los resultados obtenidos muestran que existen dos tecnologías con una calificación cualitativa de EXCELENTE, estas tecnologías son OWL y RDF cuyo valor cuantitativo es de 83.3%, mientras que la tecnología con menos capacidad de procesamiento es SWRL la misma que obtuvo una puntuación cualitativa de MUY BUENO y cuyo valor cuantitativo es del 58.3%.

### 3.3.4 Indicador 4: Programación

Para medir la programación se tomó en cuenta los parámetros:

- Familiaridad con otros lenguajes
- Tiempo de desarrollo

#### 3.3.4.1 Parámetro Familiaridad con otros Lenguajes

Para realizar la medición en cuanto a la Familiaridad con otros lenguajes se definieron los siguientes criterios de evaluación:

*Tabla 27-3. Criterios de evaluación parámetro Familiaridad con otros Lenguajes*

Valor Cuantitativo	Valor Cualitativo
1	Regular
2	Bueno
3	Muy Bueno
4	Excelente

**Realizado por:** Dalila Toledo

La medición de la familiaridad con otros lenguajes se basó en los lenguajes de programación más conocidos como son PHP, JavaScript, C# y Python. El análisis se lo realizo de la siguiente manera como se indica en la Tabla 32-3. La familiaridad con otros lenguajes se lo representa con Viñetas, teniendo en cuenta que los valores pueden ser Relacionado No relacionado, de la siguiente manera:

*Tabla 28-3. Simbolización de Familiaridad con otros Lenguajes*

Simbolización	Equivalencia
✓	Relacionado
--	No relacionado

**Realizado por:** Dalila Toledo

*Tabla 29-3. Datos obtenidos del Parámetro Familiaridad con otros Lenguajes*

	Familiaridad con otros Lenguajes				Total Familiaridad con otros Lenguajes	Valor Cuantitativo	Valor Cualitativo
	PHP	JavaScript	C#	Python			
<b>OWL</b>	✓	✓	✓	--	3	3	Muy Bueno
<b>RDF</b>	✓	✓	✓	--	3	3	Muy Bueno
<b>SWRL</b>	✓	✓	✓	--	3	3	Muy Bueno

**Realizado por:** Dalila Toledo

## **Interpretación**

El análisis de este parámetro se fundamentó tomando el criterio de la familiaridad que existe con otros lenguajes de programación como PHP, JavaScript, C# y Python.

Según los valores obtenidos se puede observar que las tres tecnologías OWL, RDF y SWRL tienen cierta relación en cuanto a los lenguajes de programación mencionados anteriormente, ya que con las tres tecnologías se puede aplicar los lenguajes de programación PHP, JavaScript y C#, teniendo en cuenta que estos lenguajes son los más utilizados en el desarrollo web.

### 3.3.4.2 Parámetro Tiempo de Desarrollo

Para la evaluación del tiempo de desarrollo se definió los siguientes criterios de evaluación:

*Tabla 30-3. Criterios de evaluación del parámetro Tiempo de Desarrollo*

Valor Cualitativo	Valor Cuantitativo	Cantidad de Horas de Desarrollo
Regular	1	>48
Bueno	2	>24 y <=48
Muy Bueno	3	>12 y <=24
Excelente	4	<=12

Realizado por: Dalila Toledo

Para la medición del tiempo de desarrollo se tomó en cuenta las horas requeridas por cada tecnología para el desarrollo de una tarea dentro una aplicación. El análisis se lo realizo de la siguiente manera como se indica en la Tabla 31-3.

*Tabla 31-3. Datos obtenidos del Parámetro Tiempo de desarrollo*

Tecnologías	Horas empleadas en el desarrollo	Valor Cuantitativo	Valor Cualitativo
OWL	21	3	Muy Bueno
RDF	19	3	Muy Bueno
SWRL	31	2	Bueno

Realizado por: Dalila Toledo

### Interpretación

De acuerdo a los valores obtenidos se observó que las tecnologías OWL y RDF brindaron un mejor tiempo de desarrollo, ya que su tiempo de desarrollo no supero las 24 horas, es por eso que se las califico con un valor cuantitativo de 3 equivalente a MUY BUENO. Por otra parte la tecnología SWRL requirió de mayor tiempo para el desarrollo de una tarea de programación, obteniendo un puntaje cuantitativo de 2 cuyo equivalente cualitativo es de BUENO.

### 3.3.4.3 Resultados del Indicador Programación

Una vez analizados los parámetros de Familiaridad con otros lenguajes y Tiempo de desarrollo, se procedió a la obtención del resultado en cuanto al índice de Programación, teniendo como criterios de evaluación los siguientes:

**Tabla 32-3. Criterios de Evaluación de la Programación**

Valor Cuantitativo	Valor Cualitativo	Porcentaje (%)
1	Regular	0% - 25%
2	Bueno	26% - 50%
3	Muy Bueno	51% - 75%
4	Excelente	76% - 100%

**Realizado por:** Dalila Toledo

Para determinar el valor cuantitativo de los datos obtenidos se aplica una regla de tres, con los siguientes datos:

El 100% son 8 puntos ya que es el valor máximo que resulta de la suma de los dos parámetros. A es el total resultante de cada tecnología.

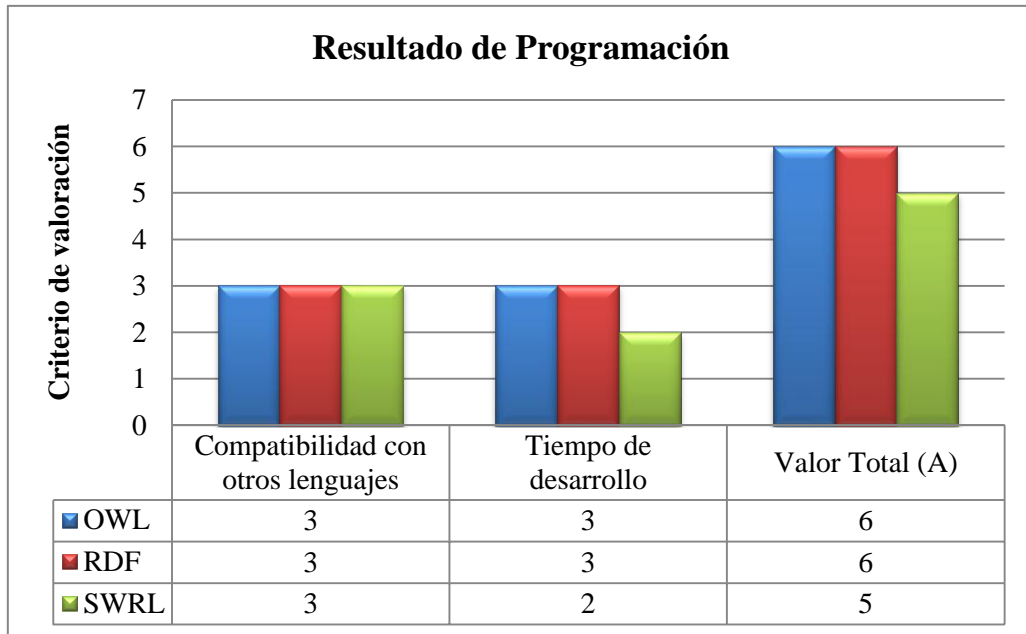
Entonces la Fórmula Utilizada es:

$$\text{Valor cuantitativo} = \frac{Ax100}{8}$$

**Tabla 33-3. Resultados obtenidos del Parámetro Programación**

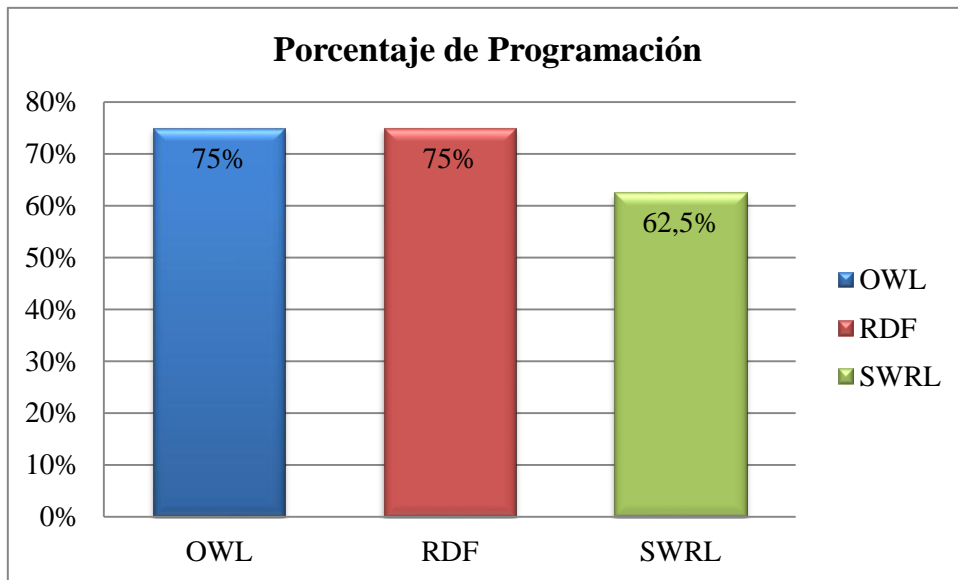
<b>Tecnologías</b>	<b>OWL</b>	<b>RDF</b>	<b>SWRL</b>
<b>Parámetros</b>			
Compatibilidad con otros lenguajes	3	3	3
Tiempo de desarrollo	3	3	2
<b>Valor Total (A)</b>	<b>6</b>	<b>6</b>	<b>5</b>
<b>Valor Cuantitativo</b>	<b>75%</b>	<b>75%</b>	<b>62.5%</b>
<b>Valor Cualitativo</b>	<b>Muy Bueno</b>	<b>Muy Bueno</b>	<b>Muy Bueno</b>

**Realizado por:** Dalila Toledo



**Figura 7-3.** Resultado del índice Programación

Realizado por: Dalila Toledo



**Figura 8-3.** Porcentajes obtenidos del índice Programación

Realizado por: Dalila Toledo



## Interpretación

Para el análisis de este indicador se tomó en cuenta la familiaridad con otros lenguajes y el tiempo de desarrollo. La familiaridad se valoró tomando en cuenta que cuando mayor es su familiaridad con otros lenguajes, mayor es la puntuación. En lo referente tiempo de desarrollo se valoró de la siguiente manera, mientras menor sea el tiempo de desarrollo más alta es su puntuación.

Por lo tanto los resultados obtenidos muestran que las tres tecnologías nos brindan una gran capacidad de programación, obteniendo una puntuación cualitativa de MUY BUENO y cuyo porcentaje es del 75% para OWL y RDF, mientras que para SWRL se obtuvo un porcentaje del 62.5%.

### 3.4 Resumen del análisis

A continuación se procedió a realizar el resumen de los resultados obtenidos durante el análisis de las tecnologías. Para una mejor comprensión se detallan las fórmulas aplicadas para la obtención de resultados.

#### 3.4.1 Fórmulas usadas para el análisis

Para obtener los resultados finales y saber cuál es la tecnología de web semántica más idónea para crear ontologías se utilizaron las siguientes fórmulas:

$$SOWL = \sum TOWL \qquad SRDF = \sum TRDF$$

$$SSWRL = \sum TSWRL \qquad ST = \sum T$$

**Dónde:**

***Towl*** = Representa el valor cuantitativo obtenido por la tecnología OWL en cada índice de comparación.

***Trdf*** = Representa el valor cuantitativo obtenido por la tecnología RDF en cada índice de comparación.

***Tswrl*** = Representa el valor cuantitativo obtenido por la tecnología SWRL en cada índice de comparación.

**Sowl** = Sumatoria de todos los valores obtenidos por la tecnología OWL en cada índice de comparación.

**Srdf** = Sumatoria de todos los valores obtenidos por la tecnología RDF en cada índice de comparación.

**Sswrl** = Sumatoria de todos los valores obtenidos por la tecnología SWRL en cada índice de comparación.

**St** = Sumatoria de los totales máximos que pueden alcanzar los valores cuantitativos de cada parámetro de comparación.

**T** = Total máximo posible de alcanzar en cada parámetro de comparación.

Porcentajes obtenidos por cada tecnología en el Análisis Comparativo:

$$Powl = \frac{Sowl}{St} * 100\%$$

$$Prdf = \frac{Srdf}{St} * 100\%$$

$$Pswrl = \frac{Sswrl}{St} * 100\%$$

### 3.4.2 Aplicación de fórmulas

Los valores obtenidos durante el análisis individual de cada índice son:

$$Sowl = 7+7+6+6=26$$

$$Srdf = .5+8+7+6=26$$

$$Sswrl = 4+5+4+5=18$$

$$St = 8+8+8+8=32$$

$$Powl = \frac{Sowl}{St} * 100\% \Rightarrow \frac{26}{32} * 100\% = 81.25\%$$

$$Prdf = \frac{Srdf}{St} * 100\% \Rightarrow \frac{26}{32} * 100\% = 81.25\%$$

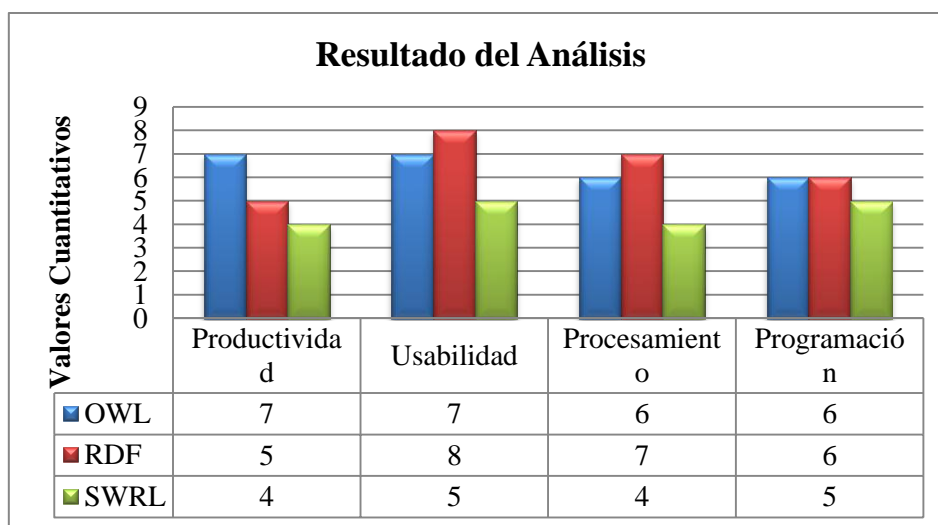
$$Pswrl = \frac{Sswrl}{St} * 100\% \Rightarrow \frac{18}{32} * 100\% = 56.25\%$$

Luego de aplicar las respectivas formulas se procedió a la obtención de los resultados finales, los mismos que se detallan a continuación:

**Tabla 34-3. Resultados Totales del Análisis Comparativo**

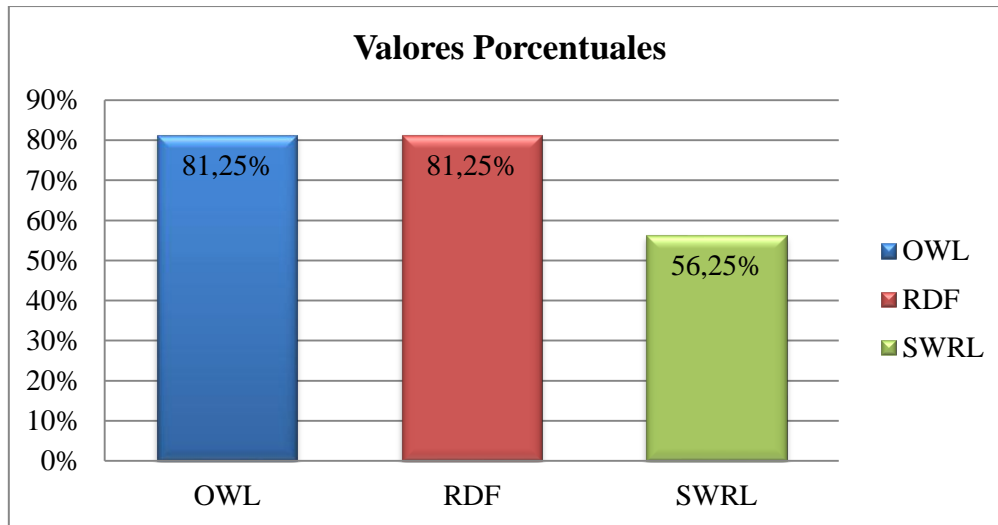
INDICADORES	VALORES CUANTITATIVOS		
	OWL	RDF	SWRL
Productividad	7	5	4
Usabilidad	7	8	5
Procesamiento	6	7	4
Programación	6	6	5
<b>TOTAL</b>	<b>(Sowl) 26</b>	<b>(Srdf) 26</b>	<b>(Sswrl) 18</b>
<b>VALORES PORCENTUALES</b> (32 = 100%)	<b>81.25%</b>	<b>81.25%</b>	<b>56.25%</b>

Realizado por: Dalila Toledo



**Figura 9-3. Resultado del análisis**

Realizado por: Dalila Toledo



**Figura 10-3.** Porcentajes Obtenidos del Análisis

Realizado por: Dalila Toledo

### 3.4.3 Interpretación de resultados

Como se puede observar en la figura 14-3 la tecnología que presenta una mayor productividad es OWL seguida de RDF y luego SWRL., Mientras que en cuanto a usabilidad la tecnología con mayor puntuación es RDF y le sigue OWL con puntuaciones de 8 y 7 respectivamente, . En el índice de procesamiento la tecnología que brinda un mayor porcentaje es RDF, con una pequeña diferencia le sigue OWL y finalmente SWRL. Por otro lado en el índice de programación las tecnologías OWL y RDF presentan una puntuación igual y la tecnología SWRL tiene una puntuación menor demostrando así que nuevamente las tecnologías OWL y RDF superan a la tecnología SWRL.

Según la figura 15-3 de los porcentajes obtenidos, se observa que las tecnologías OWL y RDF obtuvieron un porcentaje igual, demostrando así que estas tecnologías son las mejores para la creación de ontologías.

### 3.5 Comprobación de la hipótesis

Una vez realizado el respectivo análisis de cada una de las tecnologías se procedió a la comprobación de la hipótesis planteada, esto se lo describe a continuación.

### **Hipótesis planteada**

El uso de las tecnologías RDF, OWL y SWRL, permiten estructurar de forma semántica los cursos virtuales a partir de los datos extraídos de la base de datos de un LMS.

### **Comprobación**

Tomando como referencia la figura 14-3 de resultados obtenidos y la figura 15-3 de los porcentajes alcanzados se pudo observar que las tecnologías OWL y RDF son las mejores para la estructuración de las ontologías sin embargo la tecnología SWRL también permite complementar una ontología aunque de acuerdo a los índices analizados esta tecnología es la que menos se usa al momento de crear ontologías ya que requiere de mayor cantidad de recursos.

## CAPÍTULO IV

### 4. DESARROLLO DE SOFTWARE PARA CONSTRUIR LA ONTOLOGÍA DE CURSOS VIRTUALES

#### 4.1 Introducción

Este capítulo permite detallar todos los antecedentes referentes a la creación de una ontología para crear cursos virtuales a partir de un LMS de Moodle. La misma que se basó en un entorno virtual de la ESPOCH, el cual ha sido tomado como ejemplo para la aplicación de la ontología. También se detalla todas las tareas del proceso de desarrollo de la ontología y su respectiva documentación (incluidos los manuales: técnico y de usuario), además de las tareas incluidas en el posterior proceso de evaluación de la misma.

#### 4.2 Análisis

Para poder realizar el desarrollo de la ontología se requiere de la instalación de las diferentes herramientas, dentro de las cuales se encuentran el entorno virtual de Moodle que es el servidor en el cual se aplicara la ontología.

##### 4.2.1 *Instalación de herramientas*

Para el análisis de las diferentes tecnologías de la web semántica se trabajó en un entorno libre como lo es CentOS 6.5, así como también de diferentes herramientas como son:

- CentOS 6.5
- Servidor Web (http)
- Base de Datos MySQL
- Lenguaje de programación PHP
- HTML5
- Base de datos LMS Moodle 2.6
- phpmyadmin

- ARC2

#### 4.2.1.1 Instalación de CentOS 6.5

Para realizar la instalación de CentOS 6.5 se requiere de su instalador el mismo que se lo puede descargar de <http://isoredirect.centos.org/centos/6/isos/i386/>. Para la demostración requerida se procede a crear una máquina virtual, la misma que puede ser tanto en Virtual PC como en VMware Workstation, en este caso se trabajó en VMware Workstation. La instalación se realizó como cualquier sistema operativo.

#### 4.2.1.2 Instalación de servidor web

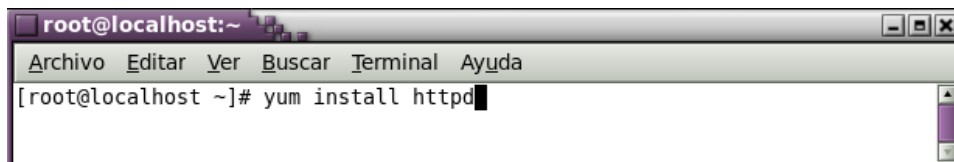
Para la instalación del servidor web primero se requiere que el equipo esté conectado a internet y que el firewall o cortafuegos este desactivado, esto lo podemos verificar ejecutando el comando:

```
# service iptables status
```

En caso que se este activo, lo procede a desactivar ejecutando el comando:

```
# service iptables stop
```

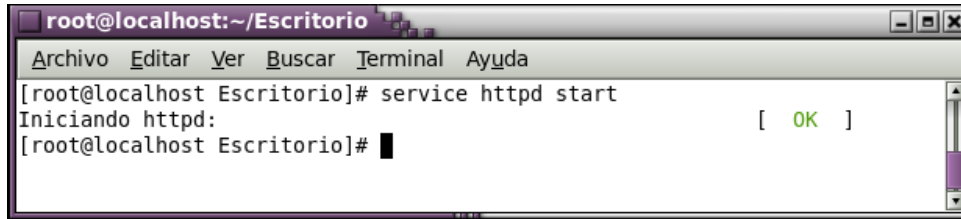
Luego procedemos a ejecutar el comando `yum install httpd` como se visualiza en la Figura 1-4. Posteriormente pedirá la confirmación para la instalación, a lo cual se debe aceptar o presionar la letra y.



**Figura 1-4.** Instalación de HTTP

**Realizado por:** Dalila Toledo

Para iniciar el servidor web se debe ejecutar el comando `service httpd start` como se visualiza en la figura 2-4.

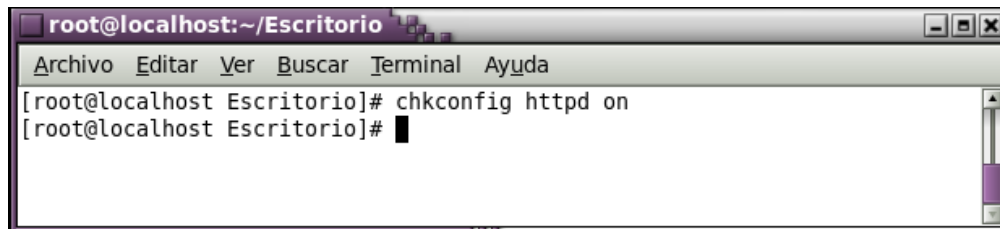


```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost Escritorio]# service httpd start
Iniciando httpd: [ OK ]
[root@localhost Escritorio]#
```

**Figura 2-4.** Inicio de Servicio Web

Realizado por: Dalila Toledo

Para completar, se configura para que el servicio web se inicie automáticamente cuando se enciende el equipo, ejecutando el comando `chkconfig httpd on` como se ve en la figura 3-4.



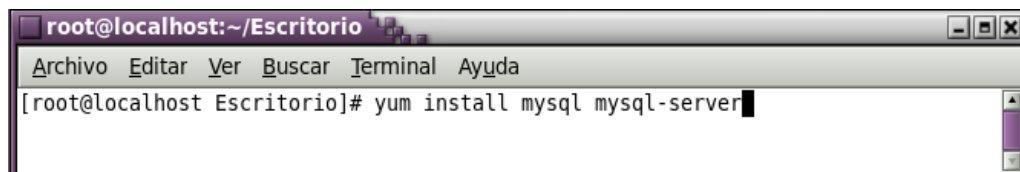
```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost Escritorio]# chkconfig httpd on
[root@localhost Escritorio]#
```

**Figura 3-4.** Inicio Automático de HTTP

Realizado por: Dalila Toledo

#### 4.2.1.3 Instalación de MySQL

Para la instalación de MySQL se requiere que el equipo esté conectado a internet, una vez verificada la conexión se procede a la instalación mediante el comando `yum install mysql mysql-server` como se ve en la Figura 4-4. Posteriormente pedirá la confirmación para la instalación, a lo cual se debe aceptar o presionar la letra `y`.



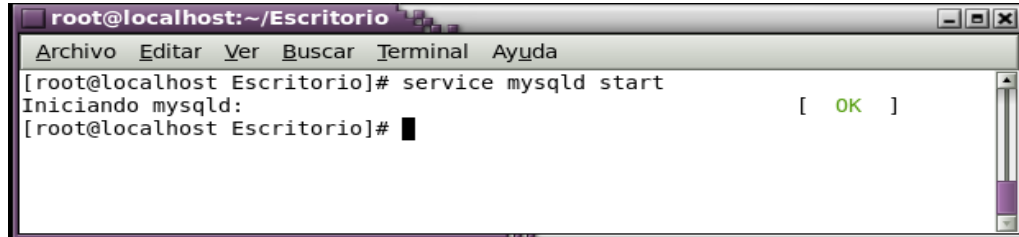
```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost Escritorio]# yum install mysql mysql-server
```

**Figura 4-4.** Instalación de MySQL

Realizado por: Dalila Toledo



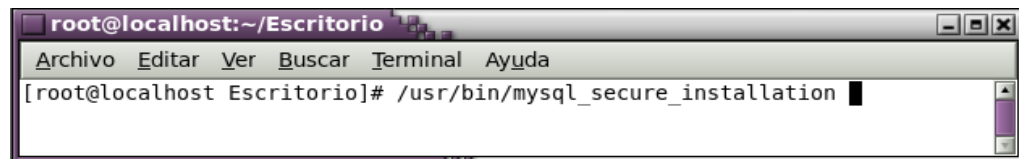
Una vez completada la instalación se debe iniciar el servicio, para lo cual se ejecuta el comando `service mysqld start` como se ve en la figura 5-4. Luego para configurar la seguridad de MySQL se debe ejecutar el comando `/usr/bin/mysql_secure_installation` como se ve en la figura 6-4.



```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost Escritorio]# service mysqld start
Iniciando mysqld: [ OK ]
[root@localhost Escritorio]#
```

**Figura 5-4.** Inicio de Servicio MySQL

Realizado por: Dalila Toledo

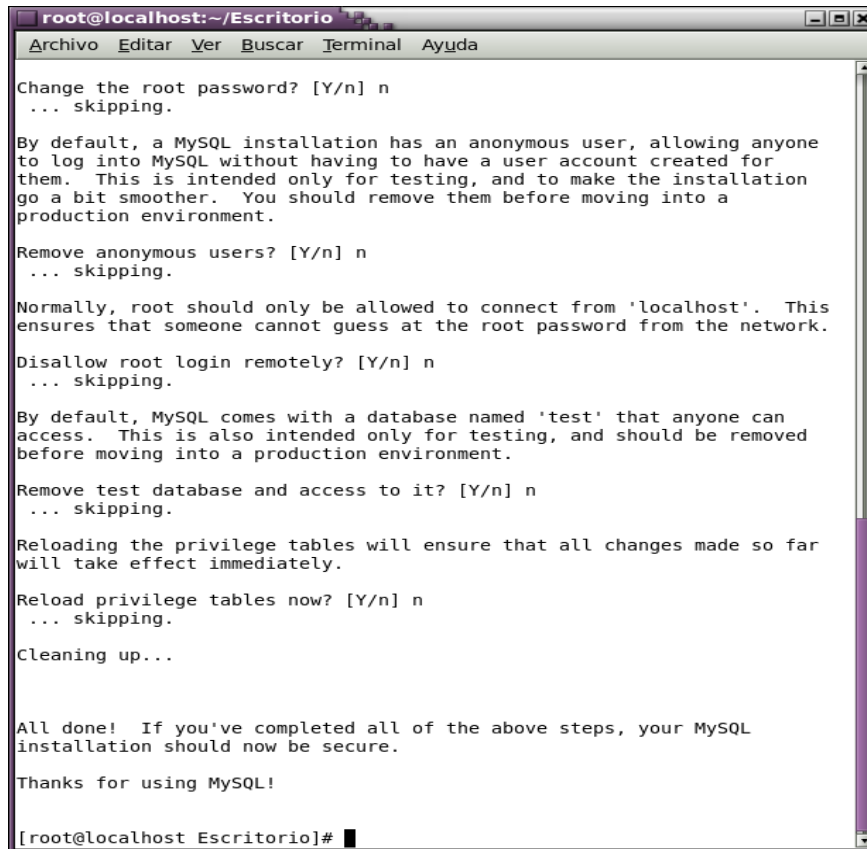


```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost Escritorio]# /usr/bin/mysql_secure_installation
```

**Figura 6-4.** Configuración de Seguridades de MySQL

Realizado por: Dalila Toledo

Aquí se podrá cambiar el password de MySQL, mover los usuarios anónimos, deshabilitar la administración remota, remover los test de acceso a la base y reasignar los privilegios de las tablas, Figura 7-4.



```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda

Change the root password? [Y/n] n
... skipping.

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] n
... skipping.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
... skipping.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] n
... skipping.

Cleaning up...

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

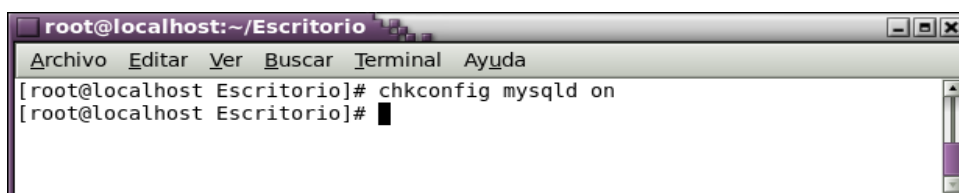
Thanks for using MySQL!

[root@localhost Escritorio]#
```

**Figura 7-4.** Seguridades MySQL

Realizado por: Dalila Toledo

Para completar, se configura para que el servicio web se inicie automáticamente cuando se enciende el equipo, ejecutando el comando `chkconfig mysqld on` como se ve en la figura 8-4.



```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda

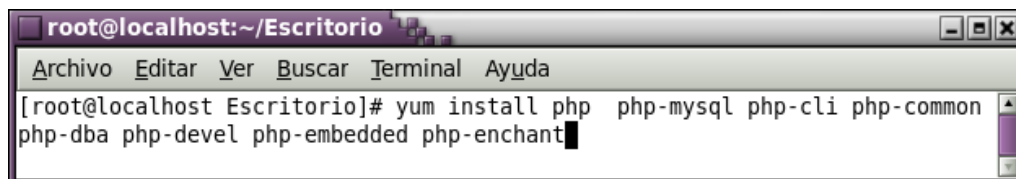
[root@localhost Escritorio]# chkconfig mysqld on
[root@localhost Escritorio]#
```

**Figura 8-4.** Inicio Automático de MySQL

Realizado por: Dalila Toledo

#### 4.2.1.4 Instalación de PHP

Para la instalación de PHP y los diferentes paquetes, se requiere que el equipo esté conectado a internet, una vez verificada la conexión se procede a la instalación mediante el comando `yum install php php-mysql php-cli php-common php-dba php-devel php-embedded php-enchant` como se ve en la Figura 9-4. Posteriormente pedirá la confirmación para la instalación de todo los paquetes, a lo cual se debe aceptar o presionar la letra `y`.



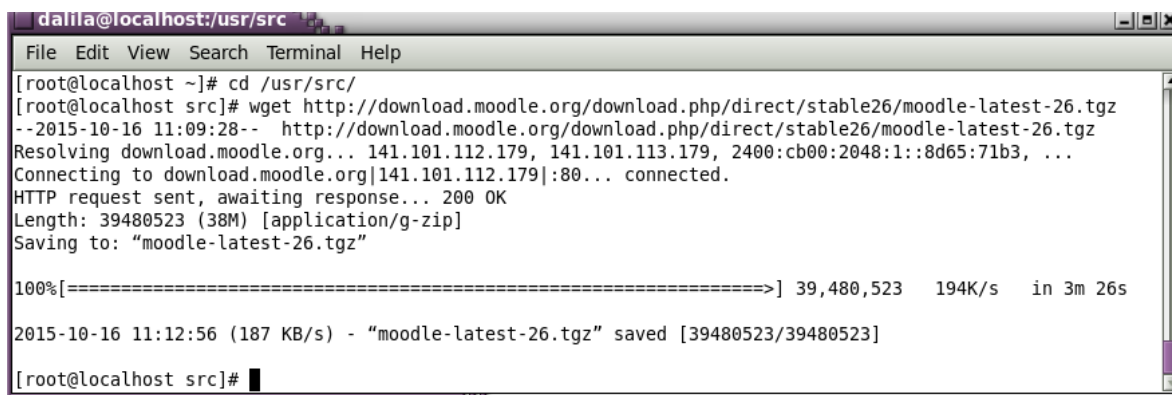
```
root@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost Escritorio]# yum install php php-mysql php-cli php-common
php-dba php-devel php-embedded php-enchant
```

**Figura 9-4.** Instalación de PHP

Realizado por: Dalila Toledo

#### 4.2.1.5 Instalación de Moodle 2.6

Para la instalación de Moodle, primeramente se debe descargar el paquete mediante el comando `wget` y la URL de descarga <http://download.moodle.org/download.php/direct/stable26/moodle-latest-26.tgz>



```
dalila@localhost:~/usr/src
File Edit View Search Terminal Help
[root@localhost ~]# cd /usr/src/
[root@localhost src]# wget http://download.moodle.org/download.php/direct/stable26/moodle-latest-26.tgz
--2015-10-16 11:09:28-- http://download.moodle.org/download.php/direct/stable26/moodle-latest-26.tgz
Resolving download.moodle.org... 141.101.112.179, 141.101.113.179, 2400:cb00:2048:1::8d65:71b3, ...
Connecting to download.moodle.org|141.101.112.179|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 39480523 (38M) [application/g-zip]
Saving to: "moodle-latest-26.tgz"

100%[=====>] 39,480,523 194K/s in 3m 26s

2015-10-16 11:12:56 (187 KB/s) - "moodle-latest-26.tgz" saved [39480523/39480523]

[root@localhost src]#
```

**Figura 10-4.** Descarga del paquete de Moodle

Realizado por: Dalila Toledo

Posteriormente se procedió a descomprimir el archivo ya que este esta con una extensión *.tgz*. Para descomprimir usamos el comando *#tar -zxvf moodle-latest-26.tgz*. como se observa en la figura 11-4

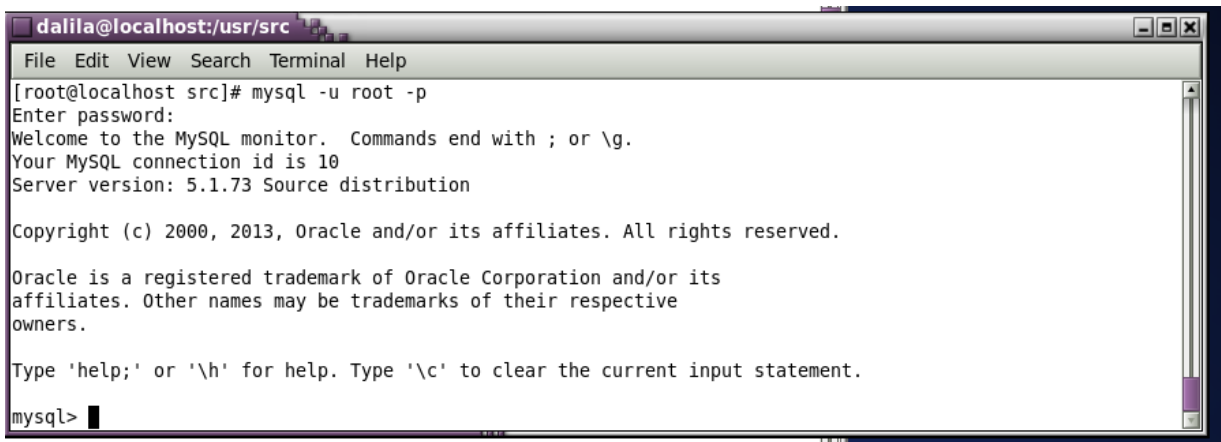


```
dalila@localhost:/usr/src
File Edit View Search Terminal Help
[root@localhost src]# tar -zxvf moodle-latest-26.tgz
```

**Figura 11-4.** Descompresión del Paquete de Moodle

Realizado por: Dalila Toledo

Una vez descomprimido podemos verificar con el comando *ls* y a continuación se procede a crear la base de datos para Moodle 2.6, usando el comando *mysql -u root*



```
dalila@localhost:/usr/src
File Edit View Search Terminal Help
[root@localhost src]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

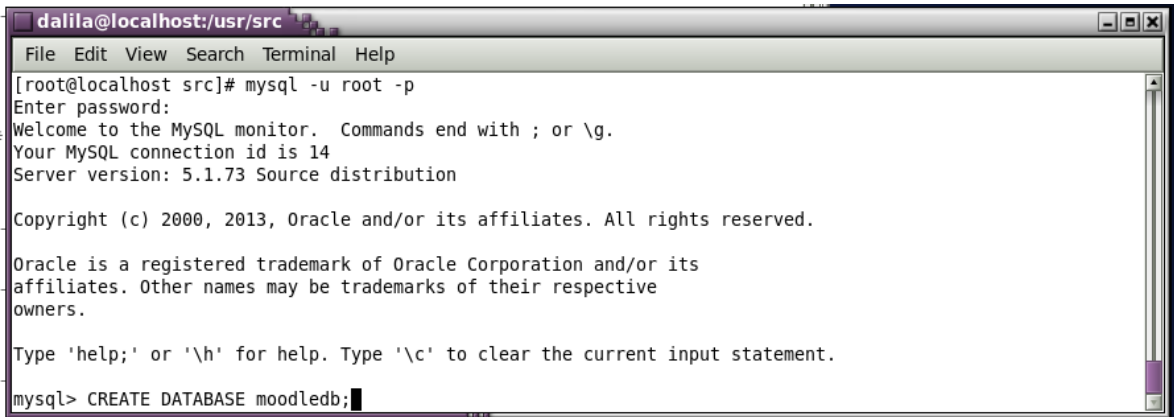
mysql>
```

**Figura 12-4.** Ingreso a la Base MySQL

Realizado por: Dalila Toledo

Y se crea la base de la siguiente manera:

Primero se ingresa a *mysql* y se ejecuta el comando *CREATE DATABASE moodledb;* como se observa en la figura 13-4.



```
dalila@localhost:/usr/src
File Edit View Search Terminal Help
[root@localhost src]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

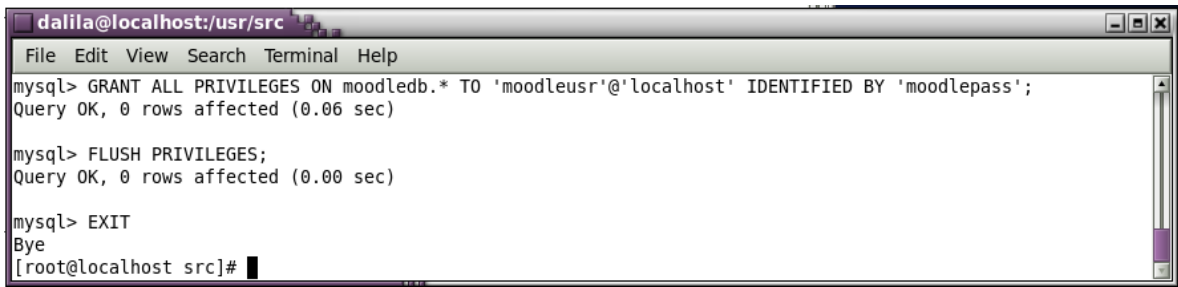
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE moodledb;
```

**Figura 13-4.** Creación de la Base de Moodle

Realizado por: Dalila Toledo

Luego se asigna todos los privilegios para la base creada.



```
dalila@localhost:/usr/src
File Edit View Search Terminal Help
mysql> GRANT ALL PRIVILEGES ON moodledb.* TO 'moodleusr'@'localhost' IDENTIFIED BY 'moodlepass';
Query OK, 0 rows affected (0.06 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> EXIT
Bye
[root@localhost src]#
```

**Figura 14-4.** Asignación de privilegios a la base de Moodle.

Realizado por: Dalila Toledo

Una vez creada la base de datos para Moodle se continua moviendo los archivos al directorio correspondiente mediante el comando `#mv /usr/src/Moodle /var/www/html/`.



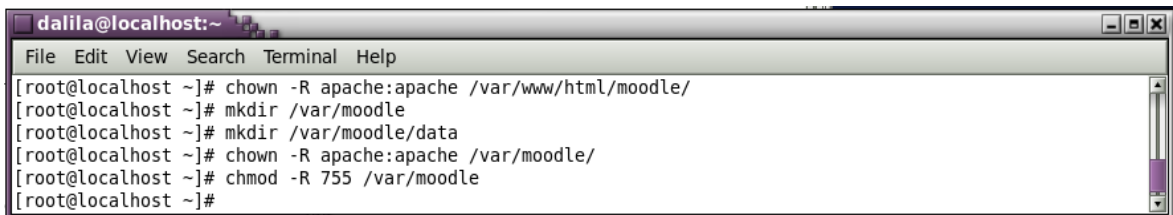
```
dalila@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# mv /usr/src/moodle /var/www/html/moodle
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
```

**Figura 15-4.** Movimiento de los archivos Descomprimidos a la Base de Moodle.

Realizado por: Dalila Toledo

Se asigna el grupo y usuario de apache mediante `#chown -R apache:apache /var/www/html/moodle/` y se crea la carpeta data en la cual se alojara los datos de Moodle, usando los siguientes comandos:

```
#mkdir /var/moodle
#mkdir /var/moodle/data
#chown -R apache:apache /var/moodle
#chmod -R 755 /var/moodle
```

A screenshot of a terminal window titled 'dalila@localhost:~'. The terminal shows a series of commands being executed in a root shell: '[root@localhost ~]# chown -R apache:apache /var/www/html/moodle/', '[root@localhost ~]# mkdir /var/moodle', '[root@localhost ~]# mkdir /var/moodle/data', '[root@localhost ~]# chown -R apache:apache /var/moodle/', and '[root@localhost ~]# chmod -R 755 /var/moodle'. The prompt returns to '[root@localhost ~]#' after each command. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'.

**Figura 16-4.** Creación de la carpeta datos de Moodle

Realizado por: Dalila Toledo

Una vez asignados los permisos necesarios para la ejecución de Moodle se procede a la configuración de los datos del servidor, para esto se ubica en la carpeta de Moodle usando el siguiente comando:

```
# cd /var/www/html/Moodle/
```

Una vez dentro del directorio de Moodle, por seguridad se realiza una copia del archivo de configuración de Moodle antes de realizar cualquier cambio, utilizando el siguiente comando:

```
# cp config-dist.php config.php
```

Luego se procede a realizar la configuración del servidor de acuerdo a los datos de la base y usuario creados anteriormente. Para poder editar el archivo de configuración de Moodle se usa el comando `# vi config.php`

```
root@localhost:/var/www/html/moodle
File Edit View Search Terminal Help
=====
// 1. DATABASE SETUP
=====
// First, you need to configure the database where all Moodle data
// will be stored. This database must already have been created
// and a username/password created to access it.
$CFG->dbtype = 'mysqli'; // 'pgsql', 'mariadb', 'mysqli', 'mssql', 'sqlsrv' or 'oci'
$CFG->dblibrary = 'native'; // 'native' only at the moment
$CFG->dbhost = 'localhost'; // eg 'localhost' or 'db.isp.com' or IP
$CFG->dbname = 'moodleb'; // database name, eg moodle
$CFG->dbuser = 'moodleusr'; // your database username
$CFG->dbpass = 'moodlepass'; // your database password
$CFG->prefix = 'mdl_'; // prefix to use for all table names
$CFG->dboptions = array(
    'dbpersist' => false, // should persistent database connections be
                        // used? set to 'false' for the most stable
                        // setting, 'true' can improve performance
                        // sometimes
    'dbsocket' => false, // should connection via UNIX socket be used?
                        // if you set it to 'true' or custom path
                        // here set dbhost to 'localhost',
                        // (please note mysql is always using socket
                        // if dbhost is 'localhost' - if you need
                        // local port connection use '127.0.0.1')
    'dbport' => '', // the TCP port number to use when connecting
                        // to the server. keep empty string for the
-- INSERT --
```

Figura 17-4. Configuración de base, nombre de usuario y password de Moodle.

Realizado por: Dalila Toledo

```
root@localhost:/var/www/html/moodle
File Edit View Search Terminal Help
=====
// 2. WEB SITE LOCATION
=====
// Now you need to tell Moodle where it is located. Specify the full
// web address to where moodle has been installed. If your web site
// is accessible via multiple URLs then choose the most natural one
// that your students would use. Do not include a trailing slash
//
// If you need both intranet and Internet access please read
// http://docs.moodle.org/en/masquerading
$CFG->wwwroot = 'http://192.168.1.10/moodle';
=====
// 3. DATA FILES LOCATION
=====
// Now you need a place where Moodle can save uploaded files. This
// directory should be readable AND WRITEABLE by the web server user
// (usually 'nobody' or 'apache'), but it should not be accessible
// directly via the web.
//
// - On hosting systems you might need to make sure that your "group" has
// no permissions at all, but that "others" have full permissions.
//
// - On Windows systems you might specify something like 'c:\moodledata'
-- INSERT --
```

Figura 18-4. Configuración de la IP del servidor de Moodle.

Realizado por: Dalila Toledo

```
root@localhost:/var/www/html/moodle
File Edit View Search Terminal Help
=====
// 3. DATA FILES LOCATION
//=====
// Now you need a place where Moodle can save uploaded files. This
// directory should be readable AND WRITEABLE by the web server user
// (usually 'nobody' or 'apache'), but it should not be accessible
// directly via the web.
//
// - On hosting systems you might need to make sure that your "group" has
// no permissions at all, but that "others" have full permissions.
//
// - On Windows systems you might specify something like 'c:\moodledata'
$CFG->dataroot = '/var/moodle/data';
//=====
// 4. DATA FILES PERMISSIONS
//=====
// The following parameter sets the permissions of new directories
// created by Moodle within the data directory. The format is in
// octal format (as used by the Unix utility chmod, for example).
// The default is usually OK, but you may want to change it to 0750
// if you are concerned about world-access to the files (you will need
// to make sure the web server process (eg Apache) can access the files.
// NOTE: the prefixed 0 is important, and don't use quotes.
-- INSERT --
```

**Figura 19-4.** Configuración de la localización de la carpeta data.

Realizado por: Dalila Toledo

#### 4.2.1.6 Instalación de *phpmyadmin*

Abrimos una consola y nos cambiamos al directorio raíz de nuestras páginas web conectadas al servidor apache, descargamos la última versión de phpMyAdmin disponible al momento.

```
# cd /var/www/html/
```

```
# wget -c https://files.phpmyadmin.net/phpMyAdmin/4.6.4/phpMyAdmin-4.6.4-all-languages.tar.gz
```

Posteriormente descomprimos el archivo con el siguiente comando o directamente damos clic derecho sobre el archivo descargado y seleccionamos extraer.

```
# tar xzvf phpMyAdmin-4.1.1-all-languages.tar.gz
```

Renombramos el directorio y cambiamos los permisos para apache



```
# mv phpMyAdmin-4.1.1-all-languages phpmyadmin  
# chown -R apache.apache /var/www/html/phpmyadmin/
```

Una vez ejecutado correctamente los comandos anteriores abrimos un browser y navegamos al URL localhost/phpmyadmin.

#### **4.2.1.7 Instalación de ARC2**

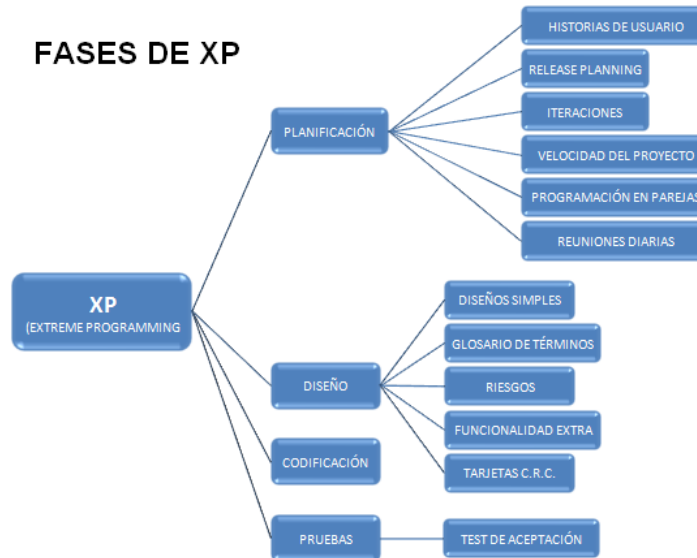
Para la instalación de ARC primeramente se debe descargar el paquete desde <https://github.com/semsol/arc2/zipball/master>.

Luego se crea una base de datos dentro de mysql para que se ejecute ARC, dicha base debe tener su respectivo usuario y contraseña. Posteriormente se procede a copiar el archivo descargado dentro de la carpeta de Moodle que es en donde se aplicó, esta carpeta está ubicada dentro del directorio */var/www/Moodle* y se descomprime el archivo, dentro del cual se encuentran varios archivos, de entre los cuales se busca el archivo *config.php*, el mismo que se lo edita y procede a configurar el nombre de la base el usuario y su respectivo password.

### **4.3 Metodología de desarrollo**

Para este proceso de desarrollo se ha optado por la utilización de una Metodología ágil para el desarrollo de aplicaciones como es la denominada XP (Extreme Programing). Esta metodología está basada en la simplicidad, la comunicación, la realimentación y reutilización del código desarrollado, con el objetivo de aumentar la productividad al desarrollo de software. Además está enfocada para el trabajo de pequeños y medianos equipos.

En la Figura 35-4 se muestra las fases de la que consta esta metodología:



**Figura 20-4.** Fases de la Metodología XP

Fuente: <http://maestria-modulo7.blogspot.com/>

### 4.3.1 *Desarrollo de la Metodología*

Una vez definida la metodología a usar se procedió al desarrollo de la misma, la cual se describe a continuación.

#### 4.3.1.1 *Planificación*

La primera fase de un proyecto de desarrollo de software es la planificación en esta fase se define el comportamiento y los requerimientos que deberá tener el sistema, todo esto se lo realiza a través de Historias de Usuario, en donde a más de definir el comportamiento que tendrán el sistema y los usuarios, se estima la velocidad del proyecto, divisiones de iteraciones y planificación de entregas del sistema.

Al final del proyecto toda la información recopilada en esta fase servirá para la realización de pruebas y verificar si se cumplieron con todos los objetivos.

#### **Especificación de Requerimientos**

En esta sección se definieron todas las características que el sistema debe cumplir.

- **Requerimientos Funcionales**

1. La aplicación deberá permitir buscar un curso en la base de datos de Moodle.
2. La aplicación deberá permitir obtener las direcciones de los recursos existentes dentro de un curso de Moodle.
3. La aplicación deberá permitir obtener la estructura de los bloques y las actividades que cada bloque posee.
4. La aplicación deberá permitir obtener la estructura de las secciones de un curso de Moodle con sus respectivos recursos y actividades.
5. La aplicación deberá permitir estructurar una ontología a partir de la información extraída en los requerimientos anteriormente mencionados.

- **Requerimientos no funcionales**

1. **Usabilidad:** La aplicación deberá tener una interfaz intuitiva y de fácil uso para el usuario.
2. **Escalabilidad:** La aplicación deberá permitir la inclusión y eliminación de nuevos recursos y funcionalidades en el futuro.
3. **Mantenibilidad:** La aplicación deberá dar la facilidad para el mantenimiento de la misma, aislar los defectos y sus posibles causas, corregir errores y atender nuevas demandas.
4. **Rendimiento:** Por ser una aplicación educativa deberá permitir tiempos de respuesta veloces.

- **Definición de Usuarios**

Un usuario es la persona que interactuará directamente con el sistema. Esta aplicación será utilizada por dos tipos de usuario en especial:

1. **Docentes:** Docentes de la ESPOCH.
2. **Técnicos:** Técnicos encargados del entorno virtual de la ESPOCH

### Historias de Usuario

Las historias de usuario son descripciones de los requerimientos estas son escritas por los usuarios sin datos técnicos sino de una manera sencilla para estimar tiempos, riesgos, responsables y descripción del proceso.

*Tabla 1-4. Historia de Usuario 1*

<b>Número: 1</b>	<b>Usuario:</b> Docente/ Técnico
<b>Nombre historia:</b> Buscar un curso en la base de datos de Moodle	
<b>Prioridad en negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en desarrollo:</b> Alto (Alto / Medio / Bajo)
<b>Iteración asignada: 1</b>	
<b>Programador responsable:</b> Dalila Toledo	
<b>Descripción:</b> El usuario podrá buscar si existe un curso dentro de la base de datos de Moodle, para de esta manera poder obtener toda la información necesaria para crear una ontología.	
<b>Observaciones:</b> <i>CONFIRMADO con el cliente</i>	

**Realizado por:** Dalila Toledo

**Tabla 2-4. Historia de Usuario 2**

<b>Número: 2</b>	<b>Usuario:</b> Docente/ Técnico
<b>Nombre historia:</b> Obtención de direcciones de los recursos	
<b>Prioridad en negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en desarrollo:</b> Alto (Alto / Medio / Bajo)
<b>Iteración asignada: 1</b>	
<b>Programador responsable:</b> Dalila Toledo	
<b>Descripción:</b> El usuario podrá visualizar las direcciones en las cuales están ubicados los recursos de un curso específico.	
<b>Observaciones:</b> CONFIRMADO con el cliente	

**Realizado por:** Dalila Toledo

**Tabla 3-4. Historia de Usuario 3**

<b>Número: 3</b>	<b>Usuario:</b> Docente/ Técnico
<b>Nombre historia:</b> Obtención de la estructura de bloques	
<b>Prioridad en negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en desarrollo:</b> Alto (Alto / Medio / Bajo)
<b>Iteración asignada: 2</b>	
<b>Programador responsable:</b> Dalila Toledo	
<b>Descripción:</b> El usuario podrá obtener la estructura de los bloques de un curso específico y sus respectivos recursos.	
<b>Observaciones:</b> CONFIRMADO con el cliente	

**Realizado por:** Dalila Toledo

**Tabla 4-4. Historia de Usuario 4**

<b>Número: 4</b>	<b>Usuario:</b> Docente/ Técnico	
<b>Nombre historia:</b> Obtención de la estructura de secciones		
<b>Prioridad en negocio:</b> Alta (Alta / Media / Baja)		<b>Riesgo en desarrollo:</b> Alto (Alto / Medio / Bajo)
<b>Iteración asignada:</b> 2		
<b>Programador responsable:</b> Dalila Toledo		
<b>Descripción:</b> El usuario podrá obtener la estructura de las secciones de un curso específico y sus respectivos recursos y actividades.		
<b>Observaciones:</b> <i>CONFIRMADO con el cliente</i>		

**Realizado por:** Dalila Toledo

**Tabla 5-4. Historia de Usuario 5**

<b>Número: 5</b>	<b>Usuario:</b> Docente/ Técnico	
<b>Nombre historia:</b> Estructuración de una ontología		
<b>Prioridad en negocio:</b> Alta (Alta / Media / Baja)		<b>Riesgo en desarrollo:</b> Alto (Alto / Medio / Bajo)
<b>Iteración asignada:</b> 3		
<b>Programador responsable:</b> Dalila Toledo		
<b>Descripción:</b> El usuario podrá estructurar una ontología a partir de los datos extraídos de la base de datos de Moodle.		
<b>Observaciones:</b> <i>CONFIRMADO con el cliente</i>		

**Realizado por:** Dalila Toledo

### **Plan de Entregas**

Una vez determinados los requisitos que debe cumplir la aplicación y las historias de usuario, se planeó un calendario de entregas entre el cliente propietario del sistema y el desarrollador, dicho plan se detalla a continuación:

*Tabla 6-4. Plan de Entregas*

<b>PLAN DE ENTREGAS</b>							
N° Historia de Usuario	Horas Estimadas	Iteración Asignada			Entrega Asignada		
		1	2	3	1	2	3
1	20	*			*		
2	20	*			*		
3	20		*			*	
4	20		*			*	
5	60			*			*
<b>TOTAL DE HORAS POR ENTREGA</b>					<b>40</b>	<b>40</b>	<b>60</b>
<b>TOTAL DE HORAS</b>					<b>140</b>		

**Realizado por:** Dalila Toledo

### **Plan de Iteraciones**

Las iteraciones se desarrollaran en un tiempo de trabajo semanal de lunes a viernes.

#### **Iteración 1**

En esta iteración se realizó la búsqueda de un curso y la obtención de los recursos de un curso de Moodle, y se cumplirá con los requerimientos funcionales 1 y 2.

*Tabla 7-4. Iteración 1*

<b>Requerimiento Funcional</b>	<b>Duración en Días</b>
1. La aplicación deberá permitir buscar un curso en la base de datos de Moodle.	1
2. La aplicación deberá permitir obtener las direcciones de los recursos existentes dentro de un curso de Moodle.	3

**Realizado por:** Dalila Toledo

## Iteración 2

En esta iteración se desarrolló la obtención de la estructura de los bloques y secciones de un curso, cumpliendo con los requerimientos funcionales 3 y 4.

*Tabla 8-4. Iteración 2*

<b>Requerimiento Funcional</b>	<b>Duración en Días</b>
3. La aplicación deberá permitir obtener la estructura de los bloques y las actividades que cada bloque posee.	2
4. La aplicación deberá permitir obtener la estructura de las secciones de un curso de Moodle con sus respectivos recursos y actividades.	3

**Realizado por:** Dalila Toledo

## Iteración 3

En la tercera iteración se llevó a cabo el requerimiento funcional número 5 el cual se trata de estructurar la ontología a partir de la información obtenida anteriormente.

*Tabla 9-4. Iteración 3*

<b>Requerimiento Funcional</b>	<b>Duración en Días</b>
5. La aplicación deberá permitir estructurar una ontología a partir de la información extraída en los requerimientos anteriormente mencionados.	10

**Realizado por:** Dalila Toledo



## Velocidad del Proyecto

Dependiendo del número de Historias de Usuario y horas a trabajar semanalmente definirán a continuación la velocidad y tiempo que llevara el proyecto.

*Tabla 10-4. Velocidad de Proyecto*

	Iteración 1	Iteración 2	Iteración 3
Horas	40	40	60
Semanas	1	1	2
Horas Semanales	40	40	60
Historias de Usuario	1 - 2	3 - 4	5

**Realizado por:** Dalila Toledo

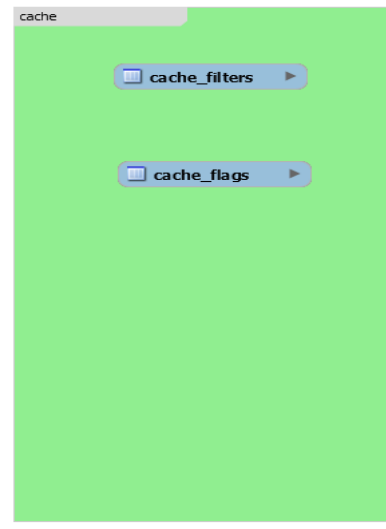
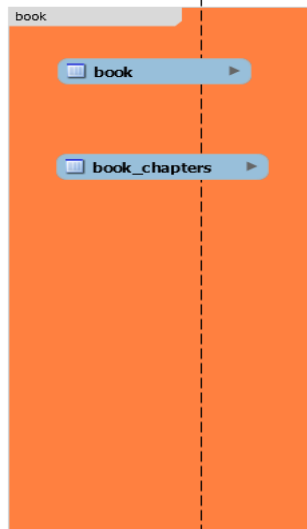
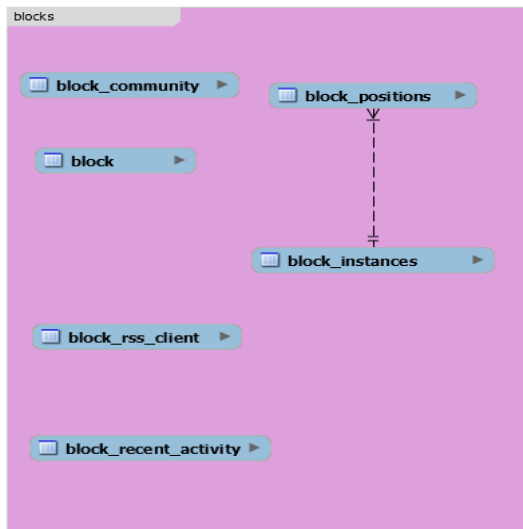
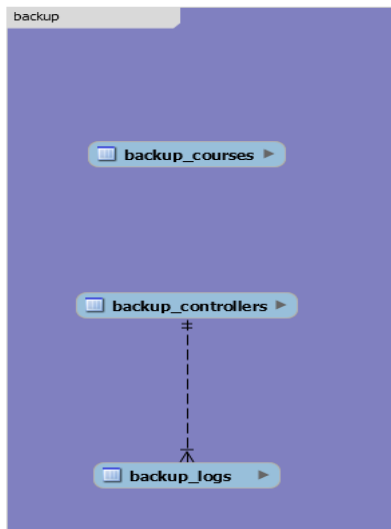
### 4.3.1.2 Diseño

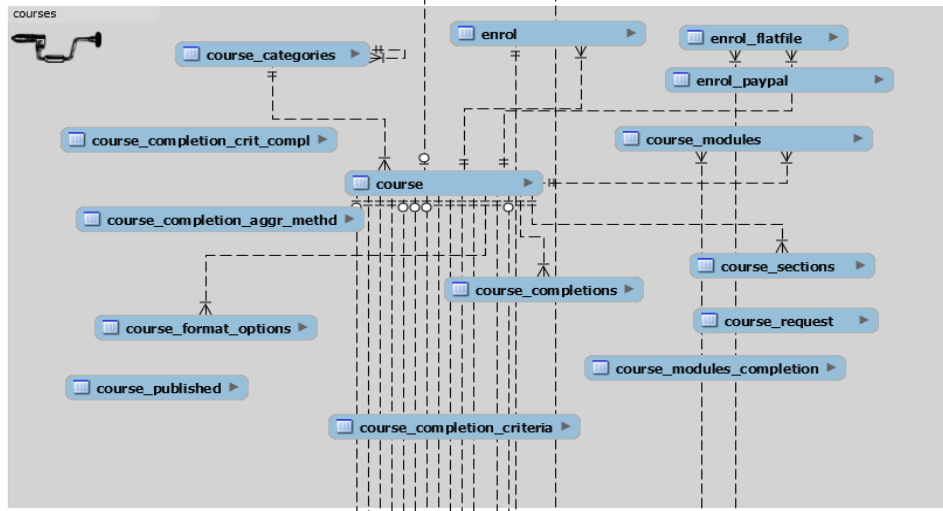
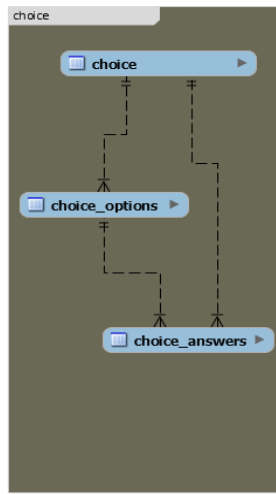
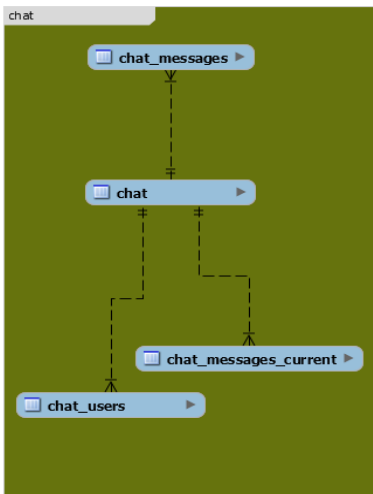
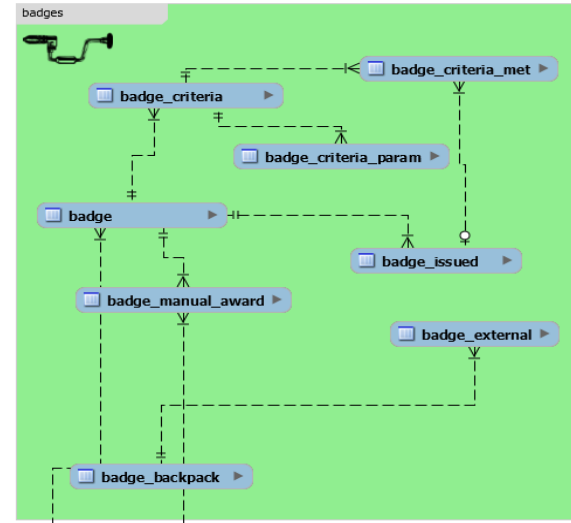
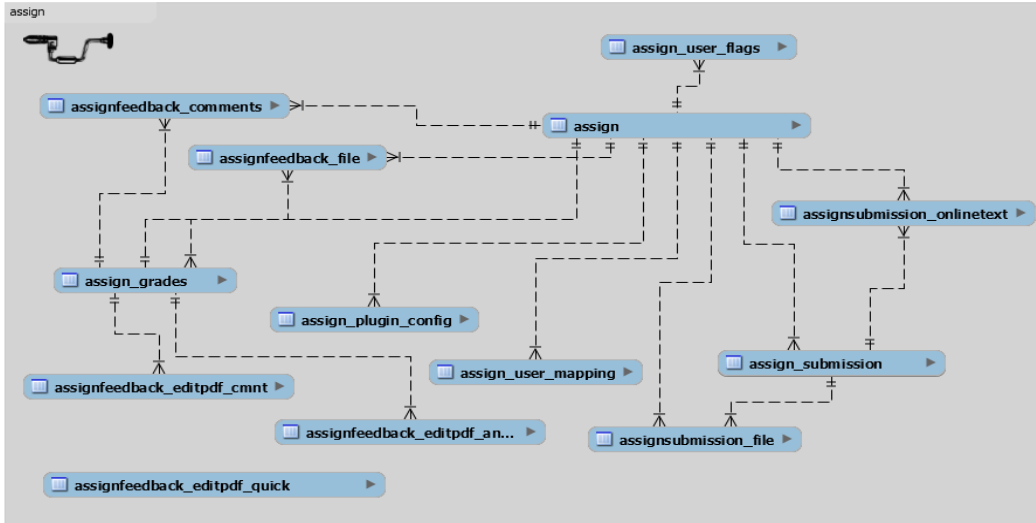
En la etapa de diseño se procede a especificar el diseño tanto de la base de datos a trabajar como de las interfaces.

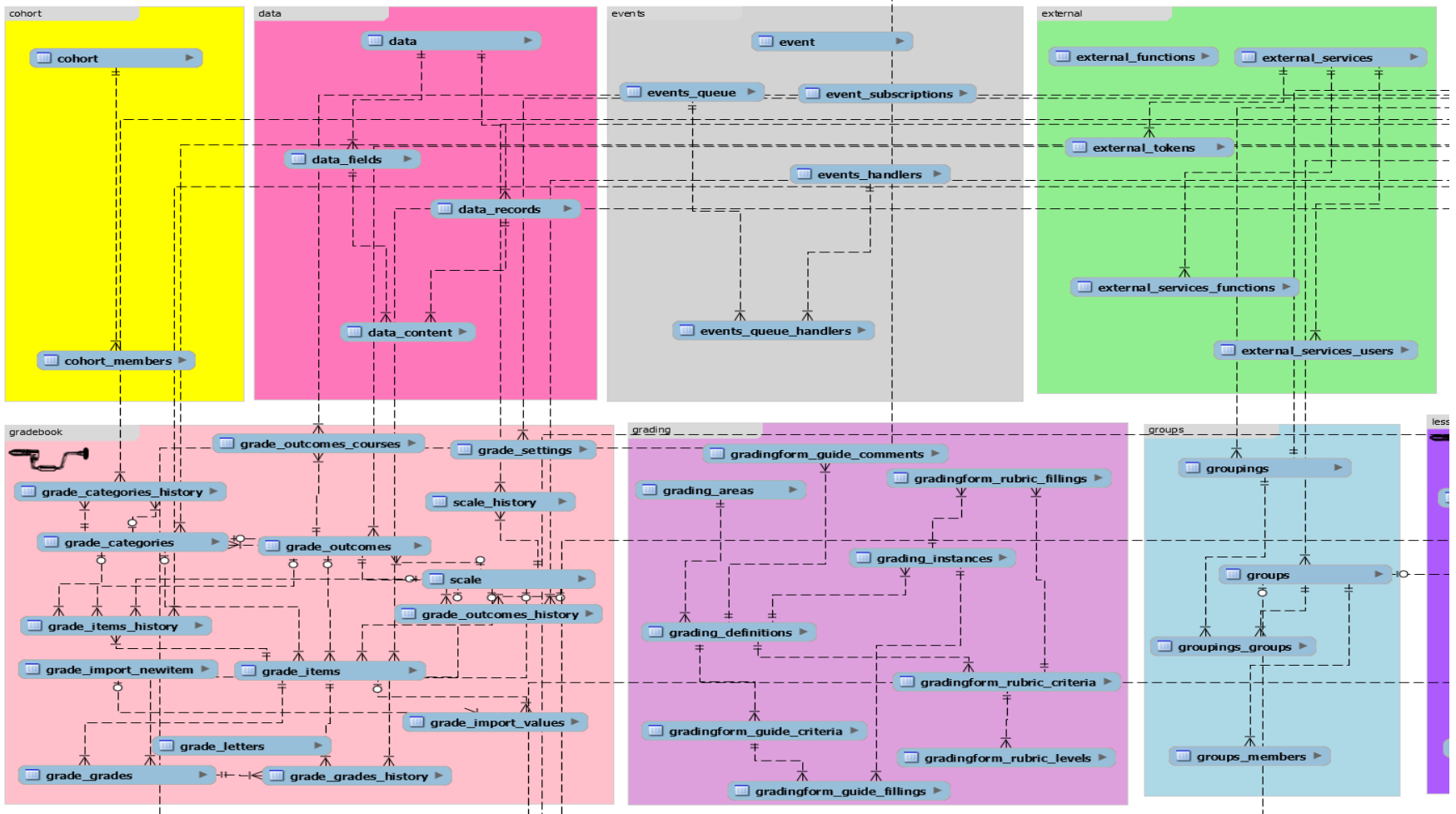
#### Base de Datos

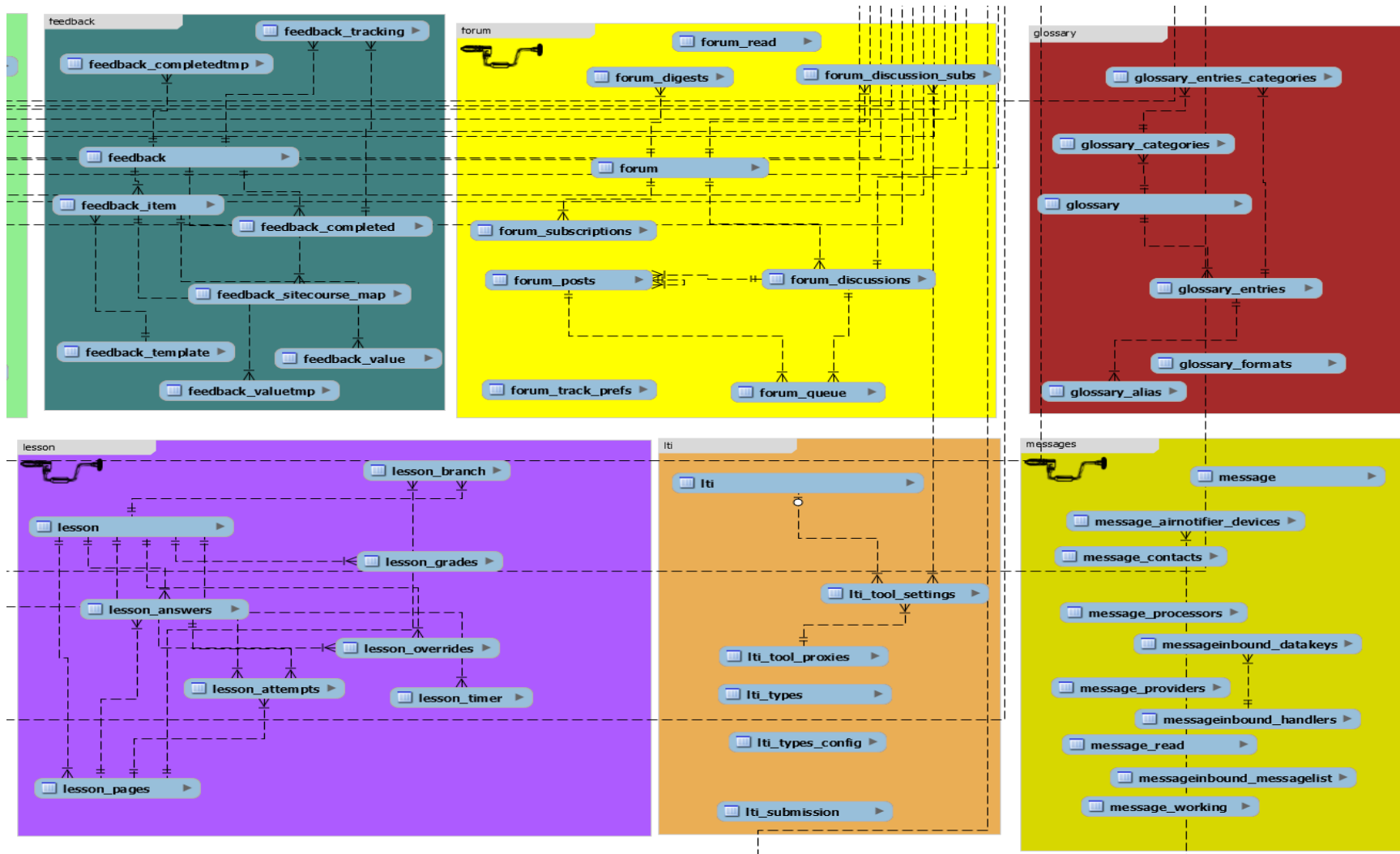
La base de datos usada para el desarrollo de la aplicación es la base de datos de Moodle, ya que a partir de la información almacenada dentro de esta, se procede a la obtención de datos para la estructuración de la ontología, la cual posteriormente permitirá la creación de cursos virtuales.

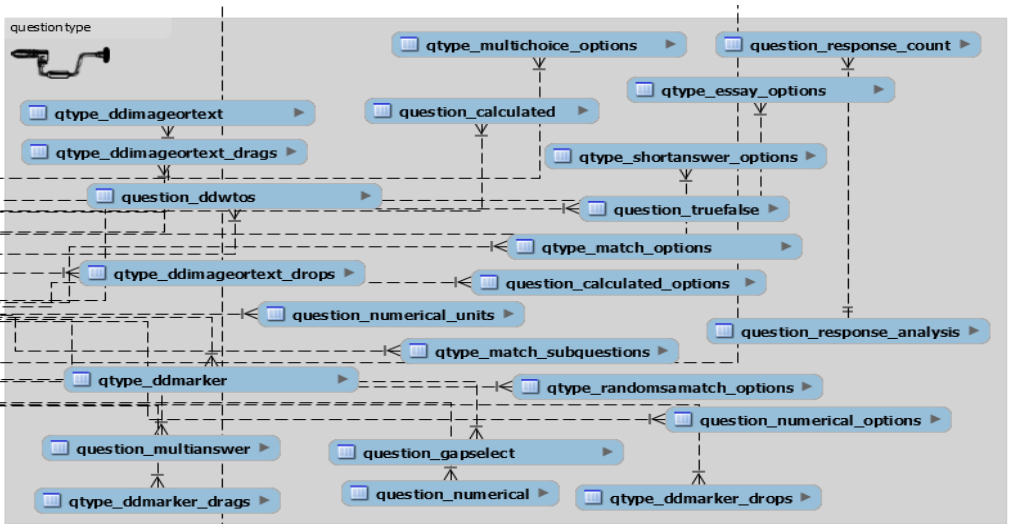
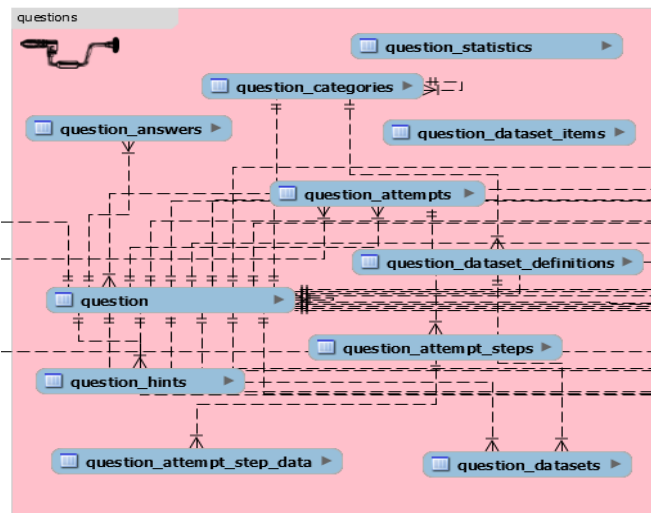
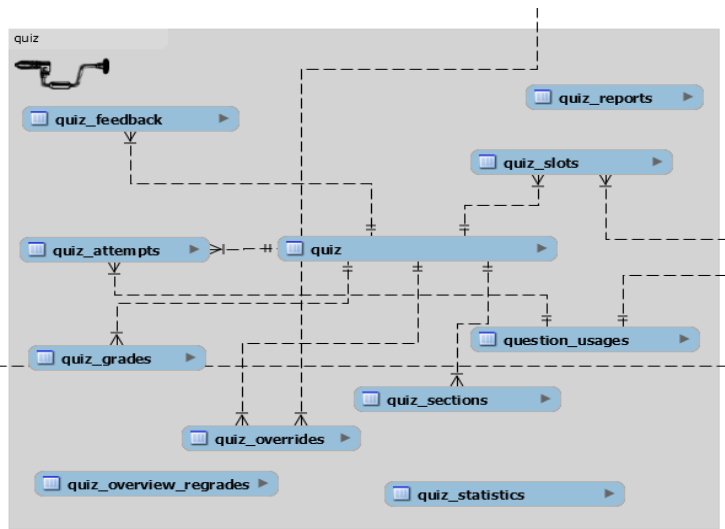
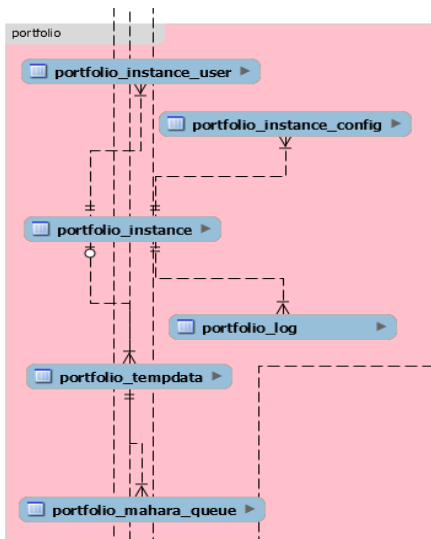
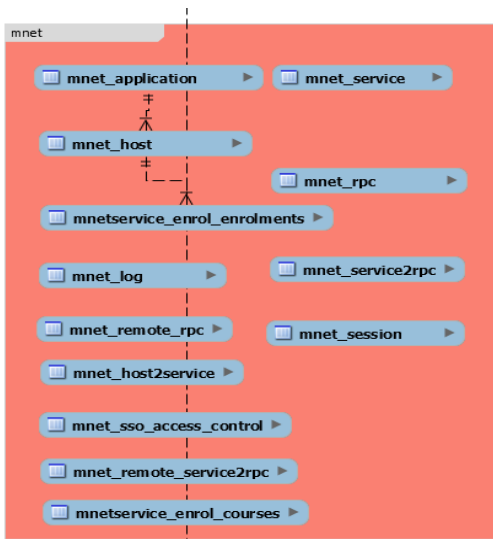
La principal información que se obtuvo de la base de Moodle es la de un curso virtual, su estructura en cuanto a bloques, secciones y sus respectivos recursos y actividades. El esquema de la base utilizada para el presente trabajo de titulación es el que se observa en la figura 36-4.

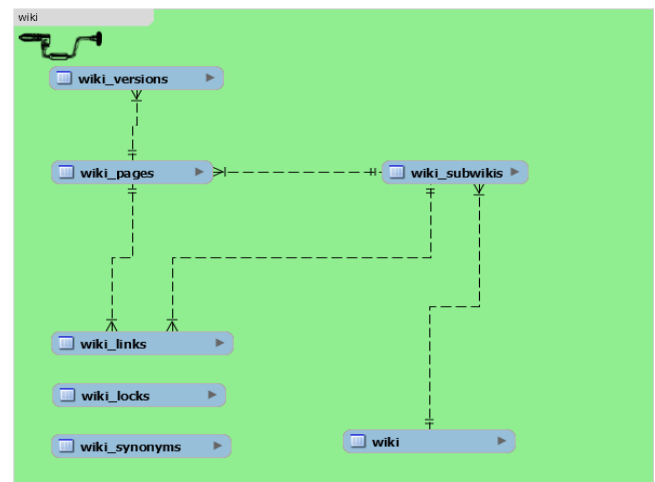
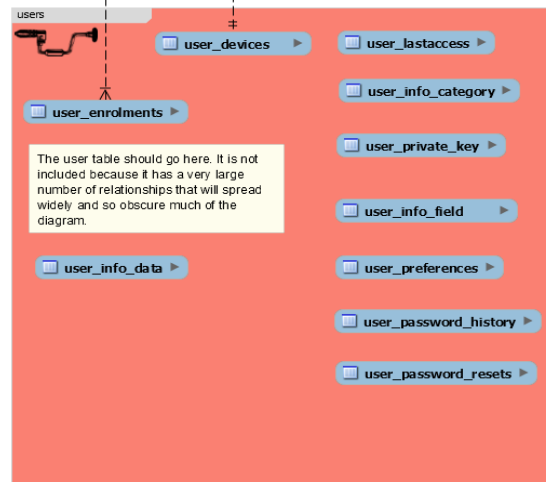
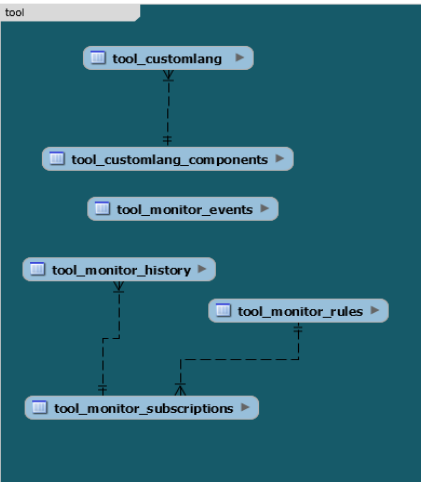
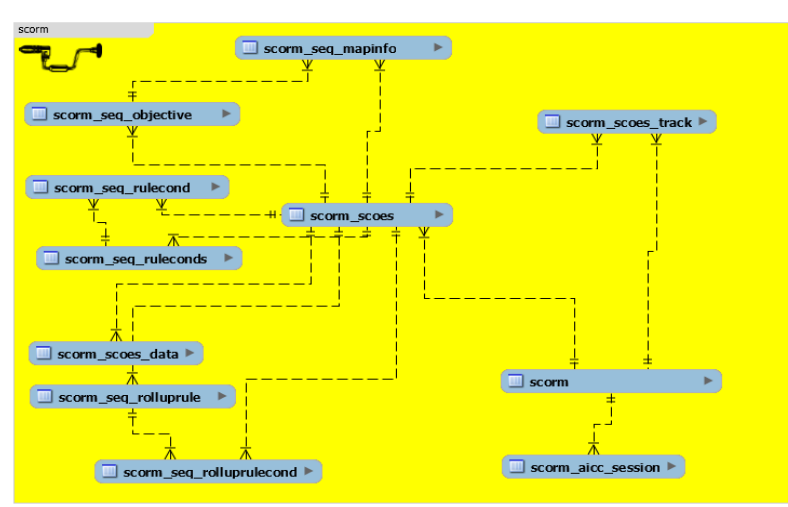
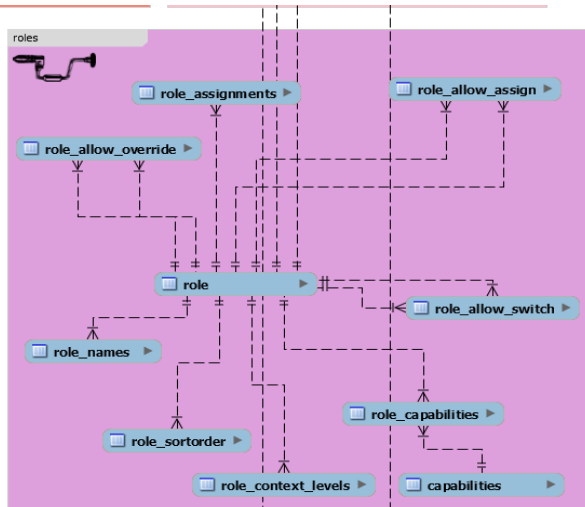
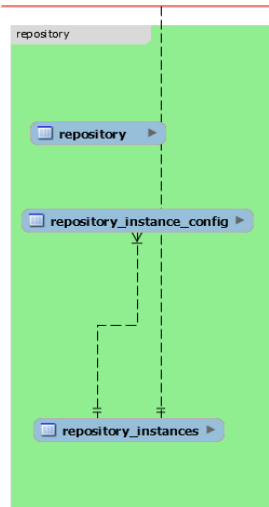












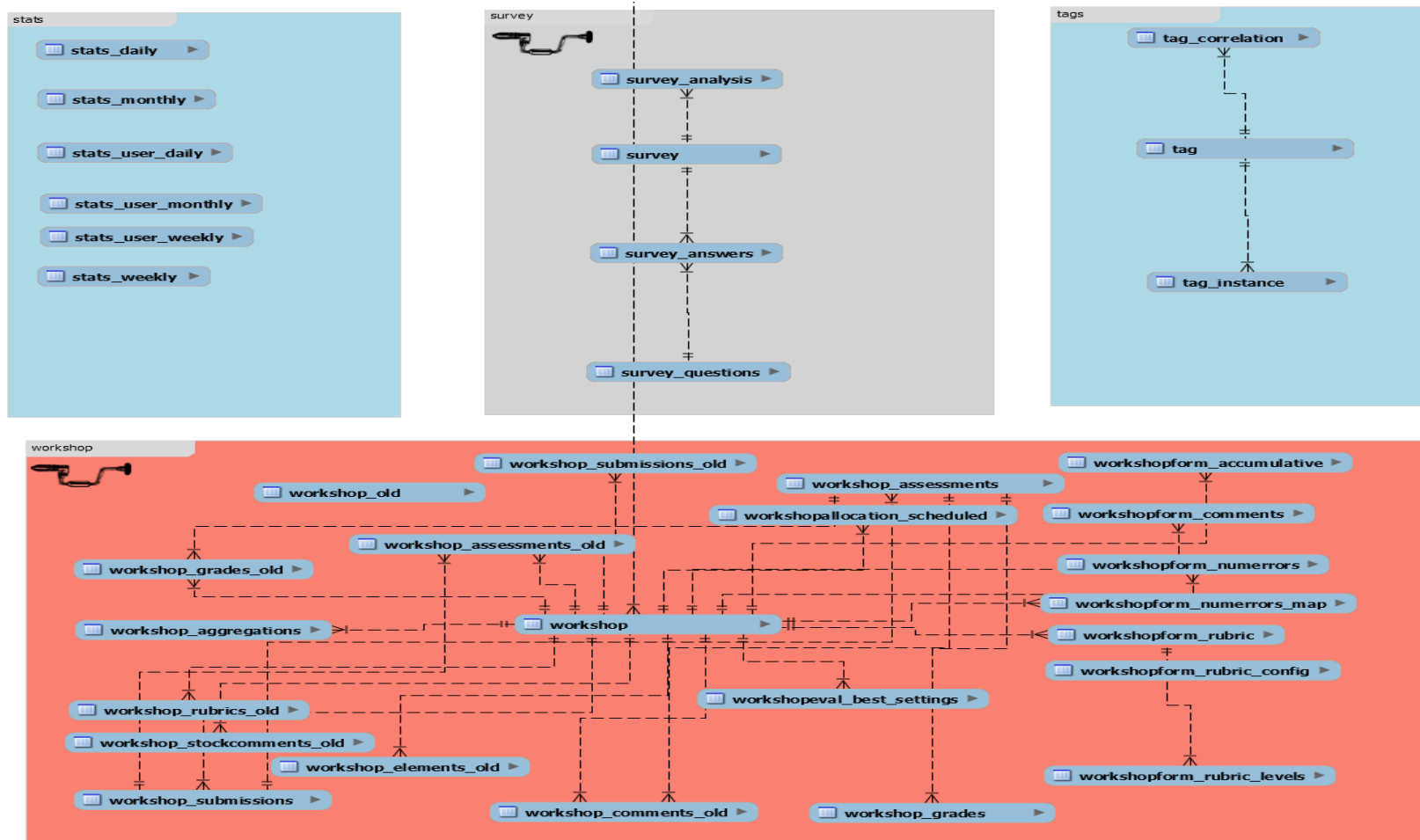


Figura 21-4. Esquema de la base de Moodle

Fuente: <http://www.examulador.com/er/>



## Diseño de Interfaces

La metodología de desarrollo XP exige que los diseños de la aplicación deban ser simples y sencillos e intuitivos, con lo que se garantizara que la implementación sea también menos complicada y en tiempos menores de desarrollo.

The wireframe shows a search interface titled "BUSCAR UN CURSO". It features a text input field with the placeholder "Que curso desea buscar..?" and a "BUSCAR" button. Below the input is a list of six courses, each with a radio button. "Curso 1" is selected. A "Ver Detalles" button is located at the bottom right.

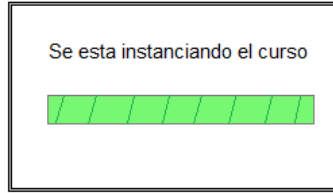
**Figura 22-4.** Diseño de aplicación para buscar un curso

Realizado por: Dalila Toledo

The wireframe displays the course details under the heading "ESTE CURSO CONSTA DE:". It lists "N BLOQUES" and "N SECCIONES". A scrollable area contains three sections: "Sección 0" with "Foro 1", "Wiki", "Presentación", and "Documento.pdf"; "Sección 1" with "Foro 2", "Documento 1", "Prueba 1", and "Foro 2"; and "Sección N" with "Documento 2", "Página 1", and "Prueba 2". "VOLVER" and "GENERAR" buttons are at the bottom.

**Figura 23-4.** Diseño aplicación que presenta la información de un curso

Realizado por: Dalila Toledo



**Figura 24-4.** Diseño mientras se genera el archivo de la ontología

**Realizado por:** Dalila Toledo

#### 4.3.1.3 Codificación

En esta fase del desarrollo de la aplicación de web semántica se describe la parte central como es el código fuente de la aplicación el cual define su comportamiento y funcionalidades.

Para la generación del archivo *.owl* se trabaja en dos partes, de las cuales la primera es estándar, aquí se encuentran definidas las propiedades y las clases de la ontología, es decir aquí se estructura la ontología de acuerdo a la lógica, y esta parte no varía entre un curso y otro. Mientras que la segunda parte que es la parte individual de la ontología es decir esta parte se genera de acuerdo a los datos que se extraigan mediante una función, esta función tiene la estructura de las tripletas de la ontología como se ve a continuación.

```
public static function getOntologia($details) {
    $ind = "";
    foreach (($details->details) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'. $element->name. '-->'
        . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'.
        $element->name .'">'
        . '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'. $element-
        >tipo .'">'
        . '</owl:NamedIndividual>';
        $ind .= $tripleta;
    } foreach (($details->details1) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'. $element->name. '-->'
        . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'.
        $element->name .'">'

```

```

.<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
>tipo ."/>'
.</owl:NamedIndividual>;
$ind .= $tripleta;
} foreach (($details->details2) as $element) {
$tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
.<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
$element->name ."'>'
.<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
>tipo ."/>'
.</owl:NamedIndividual>;
$ind .= $tripleta;
} foreach (($details->details3) as $element) {
$tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
.<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
$element->name ."'>'
.<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
>tipo ."/>'
.</owl:NamedIndividual>;
$ind .= $tripleta;
} foreach (($details->details4) as $element) {
$tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
.<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
$element->name ."'>'
.<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
>tipo ."/>'
.</owl:NamedIndividual>;
$ind .= $tripleta;
} foreach (($details->details5) as $element) {
$tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
.<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
$element->name ."'>'
.<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
>tipo ."/>'

```

```

    .'/</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details6) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.    $element-
    >tipo ."'/>'
    .'/</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details7) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    .'/<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.    $element-
    >tipo ."'/>'
    .'/</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details8) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    .'/<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.    $element-
    >tipo ."'/>'
    .'/</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details9) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    .'/<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.    $element-
    >tipo ."'/>'
    .'/</owl:NamedIndividual>';
    $ind .= $tripleta;

```

```

} foreach (($details->details10) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    . '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
    >tipo ."'/>'
    . '</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details11) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    . '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
    >tipo ."'/>'
    . '</owl:NamedIndividual>';
    $ind .= $tripleta;
}
//Obtiene cabecera de la ontologia
$pathfile = app_path()."/files/ontologia.owl";
$content = file_get_contents($pathfile);
$content = str_replace("%individuals%", $ind, $content);

//Crea y guarda el contenido de la ontologia en curso.owl
$pathfile = app_path()."/files/curso.owl";
if (is_file($pathfile)) {
    unlink($pathfile);
}
$handle = fopen($pathfile, "w+");
fwrite ($handle, $content);
fclose($handle);
}

```

Para la extracción de los detalles de los recursos de un curso se trabaja con una función, la misma que mediante sentencias sql, permite extraer los recursos de un curso específico, la función utilizada es la siguiente:

```

public static function getCourse($id) {
    $sql = "SELECT id, fullname FROM mdl_course WHERE id = ". $id;
    $header = DB::select(TRIM($sql));
    $query = "SELECT id, name, 'Recurso' as tipo FROM mdl_resource WHERE course = ". $id;
    $detail = DB::select(TRIM($query));
    $query1 = "SELECT `id`, `name`, 'Tarea' as tipo FROM `mdl_assignment` WHERE `course`
= ". $id;
    $detail1 = DB::select(TRIM($query1));
    $query2 = "SELECT i.`id`,i.`blockname` as name, 'Bloque' as tipo
        FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` =
p.`blockinstanceid`
        WHERE p.region='side-post'
        Group by i.`blockname`";
    $detail2 = DB::select(TRIM($query2));
    $query3 = "SELECT i.`id`,i.`blockname` as name, 'Bloque' as tipo
        FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` =
p.`blockinstanceid`
        WHERE p.region='side-pre'
        Group by i.`blockname`";
    $detail3 = DB::select(TRIM($query3));
    $query4 = "SELECT `id`, `name`, 'Libro' as tipo FROM `mdl_book` WHERE `course` = ".
$id;
    $detail4 = DB::select(TRIM($query4));
    $query5 = "SELECT id, section, 'Seccion' as tipo FROM `mdl_course_sections` WHERE
`course` = ". $id. " and `summary` <>''";
    $detail5 = DB::select(TRIM($query5));
    $query6 = "SELECT `id`, `name`, 'Foro' as tipo FROM `mdl_forum` WHERE `course` = ".
$id;
    $detail6 = DB::select(TRIM($query6));
    $query7 = "SELECT `id`, `name`, 'Etiqueta' as tipo FROM `mdl_label` WHERE `course` = ".
$id;
    $detail7 = DB::select(TRIM($query7));

```

```

$query8 = "SELECT `id`, `name`, 'Pagina' as tipo FROM `mdl_page` WHERE `course` = ".
$id;
$detail8 = DB::select(TRIM($query8));
$query9 = "SELECT `id`, `name`, 'Prueba' as tipo FROM `mdl_quiz` WHERE `course` = ".
$id;
$detail9 = DB::select(TRIM($query9));
$query10 = "SELECT `id`, `name`, 'Url' as tipo FROM `mdl_url` WHERE `course` = ". $id;
$detail10 = DB::select(TRIM($query10));
$query11 = "SELECT `id`, `name`, 'Wiki' as tipo FROM `mdl_wiki` WHERE `course` = ". $id;
$detail11 = DB::select(TRIM($query11));
$result = [
    'title' => $header[0],
    'details' => $detail,
    'details1' => $detail1,
    'details2' => $detail2,
    'details3' => $detail3,
    'details4' => $detail4,
    'details5' => $detail5,
    'details6' => $detail6,
    'details7' => $detail7,
    'details8' => $detail8,
    'details9' => $detail9,
    'details10' => $detail10,
    'details11' => $detail11    ];
$curso = (object)$result;
return $curso;
}

```

## Herramientas

Para esta tarea se ha utilizado las siguientes herramientas de desarrollo de software:

Tabla 11-4. Recursos utilizados para el desarrollo

HERRAMIENTA	TAREA
Netbeans	Editor de código
ARC2	Generador de archivo rdf.
Lenguaje de Programación	PHP

Realizado por: Dalila Toledo

## Base de Datos

Para el uso de la base de datos por parte de la aplicación se ha creado un archivo de tipo owl que contiene la información de la base de datos de los cursos, en donde a continuación se visualiza un ejemplo de la estructura del archivo .owl

```

- <rdf:RDF xml:base="http://localhost/owl/CourseLMS1.owl">
- <owl:Ontology rdf:about="http://localhost/owl/CourseLMS1.owl">
  <rdfs:comment>Representation of courses for LMS </rdfs:comment>
  <owl:Ontology>
- <!--
  //
  // Object Properties
  //
  //
  //
  -->
  <!-- http://localhost/owl/CourseLMS1.owl#esParteDe -->
  <owl:ObjectProperty rdf:about="http://localhost/owl/CourseLMS1.owl#esParteDe"/>
- <!--
  http://localhost/owl/CourseLMS1.owl#estaCompuestoDe
  -->
  <owl:ObjectProperty rdf:about="http://localhost/owl/CourseLMS1.owl#estaCompuestoDe">
    <owl:inverseOf rdf:resource="http://localhost/owl/CourseLMS1.owl#esparteDe"/>
  <rdfs:range>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://localhost/owl/CourseLMS1.owl#estaCompuestoDe"/>
      <owl:allValuesFrom rdf:resource="http://localhost/owl/CourseLMS1.owl#CabeceraSeccion"/>
    </owl:Restriction>
  </rdfs:range>
  </owl:Restriction>
  </rdfs:domain>
  <owl:Restriction>
  
```

Figura 25-4. Ejemplo archivo .owl

Realizado por: Dalila Toledo



#### 4.3.1.4 Pruebas

Para la validación del correcto funcionamiento de la aplicación se procede a la realización de las respectivas pruebas de cada requerimiento funcional. Esto se lo detalla a continuación:

#### **REQ 1. Buscar un curso en la base de datos de Moodle**

*Tabla 12-4. Prueba Req. 1*

Descripción	Autor
Pruebas	Dalila Toledo Ing. Danilo Pastor
Modificaciones	Dalila Toledo

**Realizado por:** Dalila Toledo

#### **Descripción**

En este requerimiento es necesario presentar todos los cursos cuyo nombre tenga relación con lo buscado.

#### **Entrada**

Se ingresa el nombre del curso que desea buscar, en este caso se usa una cadena tipo String.

#### **Resultado Deseado**

El resultado que se desea obtener es un listado de los cursos que coincidan con el nombre buscado.

#### **Resultado de Prueba**

Listado de cursos que contienen la cadena String ingresada.

#### **REQ2. Obtener las direcciones de los recursos existentes dentro de un curso de Moodle.**

*Tabla 13-4. Prueba Req. 2*

Descripción	Autor
Pruebas	Dalila Toledo Ing. Danilo Pastor
Modificaciones	Dalila Toledo

**Realizado por:** Dalila Toledo

### **Descripción**

Para este requerimiento es necesario anteriormente seleccionar un curso, a partir del cual se procede a buscar la información de los recursos que este posee.

### **Entrada**

Se envía como parámetro el código del curso seleccionado a partir de un listado de cursos presentados.

### **Resultado Deseado**

Listado de los recursos que posee el curso seleccionado.

### **Resultado de Prueba**

Se presenta de forma automática los recursos del curso seleccionado.

### **REQ3. Obtener la estructura de los bloques y las actividades que cada bloque posee.**

*Tabla 14-4. Prueba Req. 3*

Descripción	Autor
Pruebas	Dalila Toledo Ing. Danilo Pastor
Modificaciones	Dalila Toledo

**Realizado por:** Dalila Toledo

### **Descripción**

Para cumplir con este requerimiento es necesario seleccionar un curso a partir del cual se obtiene su información acerca de actividades y bloques.

### **Entrada**

Se envía como parámetro el código del curso seleccionado a partir de un listado de cursos presentados.

### **Resultado Deseado**

Listado de los bloques y actividades del curso

### **Resultado de Prueba**

Se presenta un listado de los bloques y actividades del curso seleccionado anteriormente.

**REQ4. Obtener la estructura de las secciones de un curso de Moodle con sus respectivos recursos y actividades.**

*Tabla 15-4. Prueba Req. 4*

Descripción	Autor
Pruebas	Dalila Toledo Ing. Danilo Pastor
Modificaciones	Dalila Toledo

**Realizado por:** Dalila Toledo

**Descripción**

Presentar los recursos que posee un curso en cada sección con sus respectivos recursos y actividades como tareas, chats, o foros.

**Entrada**

Se envía como parámetro el código del curso seleccionado a partir de un listado de cursos presentados.

**Resultado Deseado**

Listado detallado de los recursos y actividades de cada sección del curso.

**Resultado de Prueba**

Se presenta un listado detallado de los recursos y actividades de cada sección del curso.

**REQ5. Estructurar una ontología a partir de la información extraída en los requerimientos anteriormente mencionados.**

*Tabla 16-4. Prueba Req. 5*

Descripción	Autor
Pruebas	Dalila Toledo Ing. Danilo Pastor
Modificaciones	Dalila Toledo

**Realizado por:** Dalila Toledo

**Descripción**

Una vez obtenida toda la información de un curso específico, se genera la ontología del mismo, la cual se presenta como un archivo descargable.

**Entrada**

Se envía como parámetro el código de un curso seleccionado a partir de un listado de cursos.

**Resultado Deseado**

Generación de un archivo que posea la ontología con los datos de un curso.

**Resultado de Prueba**

Se presenta un archivo descargable con el código generado de la ontología con los datos de un curso.

## CAPÍTULO V

### 5. RESULTADOS FINALES

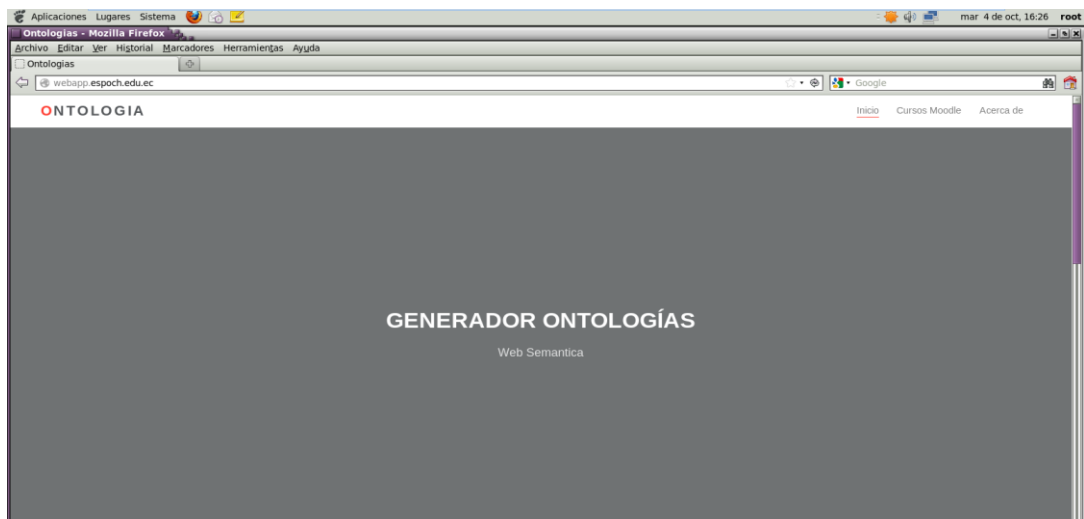
#### 5.1 Introducción

En este capítulo se presentan los resultados obtenidos a partir del análisis realizado anteriormente en el capítulo 3. Aquí se encuentra detallada la extracción de información de los cursos almacenados en la base de datos de Moodle.

#### 5.2 Demostración de resultados

Para realizar la demostración de los resultados obtenidos luego del análisis y desarrollo de la aplicación que extrae datos a partir de la información almacenada en una base de datos de Moodle, a continuación se presenta el funcionamiento de mencionada aplicación.

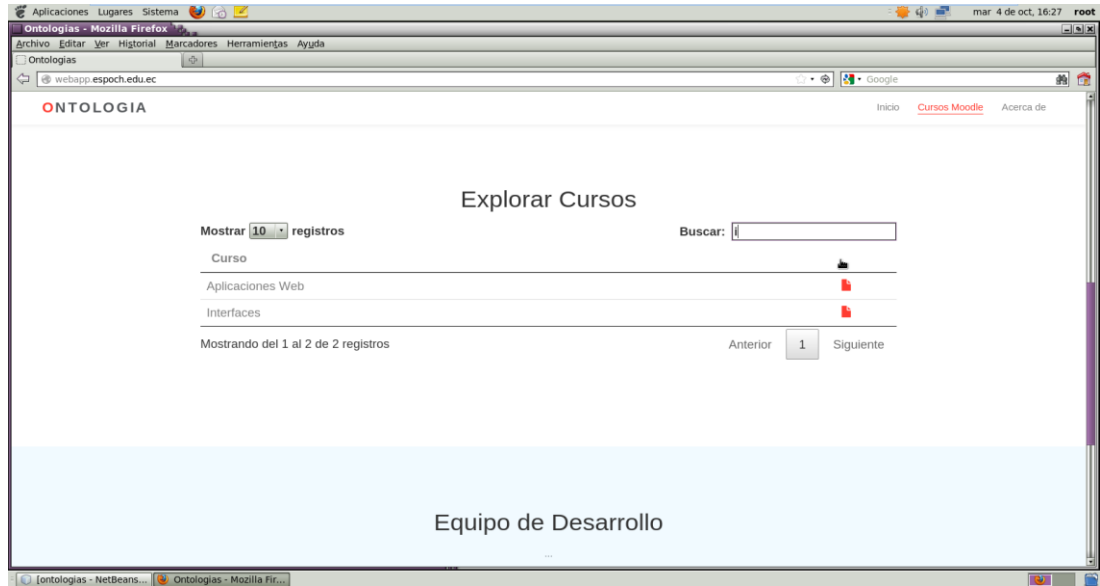
- a. **Inicio de la aplicación:** Esta aplicación puede ser ejecutada por técnicos y maestros de la ESPOCH mediante su conexión de la siguiente manera:



**Figura 1-5.** Inicio de Aplicación

Realizado por: Dalila Toledo

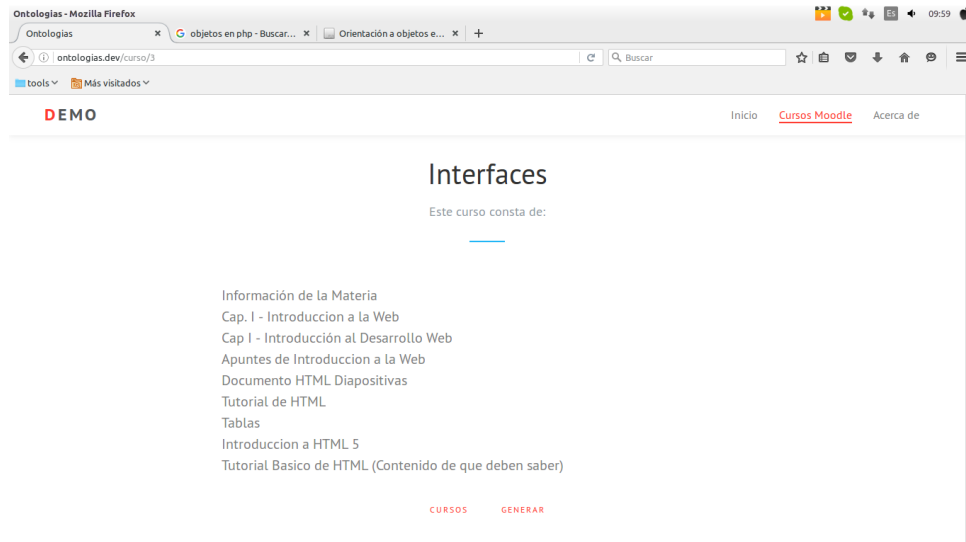
- b. **Búsqueda de información:** El usuario deberá ubicarse en la parte de *Cursos Moodle*, en donde ingresara el nombre del curso que desea buscar para que posteriormente la aplicación extraiga los cursos cuyos nombres tengan coincidencias con lo solicitado por el usuario.



**Figura 2-5.** Búsqueda de Cursos

Realizado por: Dalila Toledo

- c. **Presentación de resultados:** Una vez realizada la búsqueda, la aplicación presenta un listado de las coincidencias, a partir de las cuales el usuario puede elegir un curso para ver sus contenidos.



**Figura 3-5.** Presentación de Resultados

Realizado por: Dalila Toledo

d. **Comprobación de resultados:** La información presentada por la aplicación coincide con la información almacenada en la base de daos de Moodle.

En el siguiente ejemplo se visualiza que los recursos obtenidos del curso mediante la aplicación, concuerdan con los recursos del curso almacenado en la base de Moodle.

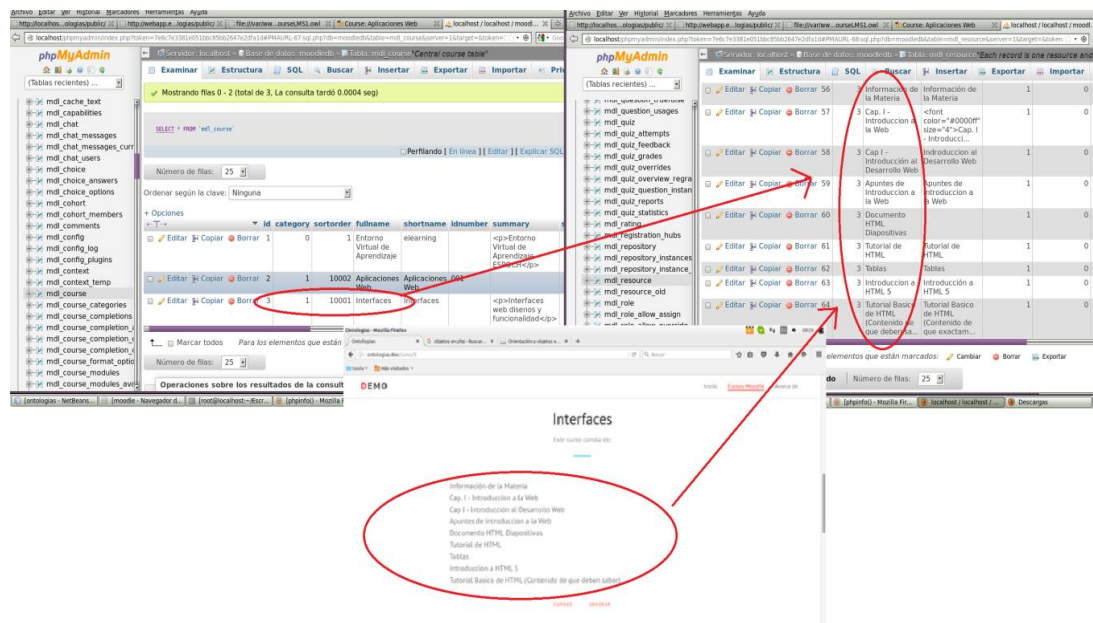


Figura 4-5. Comprobación de datos obtenidos

Realizado por: Dalila Toledo

### 5.3 Conclusión de Resultados

Luego de haber realizado el análisis de cada una de las tecnologías de la web semántica y basándonos en los índices y parámetros establecidos, se puede concluir que el uso de las tecnologías de web semántica, como RDF y OWL permite la generación de ontologías a partir de los datos extraídos de una base de datos de un LMS. Mientras que la tecnología SWRL también lo permite pero para el presente uso no se utilizan reglas.

Por lo tanto al aplicar las tecnologías de web semántica OWL y RDF, la extracción de información de los cursos virtuales almacenados en la base de datos de Moodle permite la generación de una ontología.

## CONCLUSIONES

- Las tecnologías de la web semántica y la estructura de las bases de datos de un LMS Moodle se pueden vincular de tal forma que permiten generar una ontología mediante la extracción de datos de la base.
- Luego del respectivo análisis de las tecnologías de la web semántica se pudo concluir que las tecnologías OWL y RDF permiten crear ontologías y extraer datos a partir de un entorno LMS libre.
- El desarrollo de un módulo de software que permite construir una ontología a partir de la información de base de datos del LMS es posible, mediante el uso de herramientas como ARC y el lenguaje de programación php. Este módulo es relativamente complejo ya que su estructura depende de la ontología construida.
- Para la validación de la extracción de los datos del LMS Moodle hacia la ontología estructurada en un entorno virtual, se trabajó con consultas directas a la base de Moodle, las mismas que permiten la extracción segura de los datos almacenados.
- En el análisis comparativo se obtuvo resultados de SWRL, pero en el presente trabajo esto no aplica para combinar con una ontología.
- La adaptación de la librería de ARC con Moodle es un proceso factible, pero hay que saber manejar la librería.
- Los algoritmos diseñados para generar la ontología son basados en librerías de ARC y de diseñan de acuerdo a las relaciones de los datos requeridos.



## RECOMENDACIONES

- Es importante que los contenidos de tipo educativos presentes en un curso sean creados con el apoyo de personas especialistas en pedagogía para evaluar la calidad de los contenidos y su nivel de influencia en el desarrollo de habilidades, competencias y destrezas de los estudiantes.
- La web semántica es un recurso tecnológico con un amplio campo aplicativo dentro de la educación, siendo uno de los menos explorados debido a su tiempo de investigación.
- Es necesario establecer o crear una metodología efectiva para el desarrollo de aplicaciones para asegurar la calidad del producto final, debido a que las metodologías tradicionales para desarrollo ágil utilizadas para este tipo de proyectos no toman en cuenta aspectos importantes que influyen en las aplicaciones como su constante y acelerada evolución en hardware y software, compatibilidad con varios sistemas operativos, compatibilidad con otros sistemas o aplicaciones, actualización e innovación, escenario o propósito de la aplicación y su cantidad de usuarios.
- Con el propósito de llegar a más niveles educativos con la web semántica es necesario implementar recursos educativos diversos y de complejidad evolutiva para estimular las competencias en los estudiantes.
- Se debe analizar otras alternativas de librerías diferentes a ARC para trabajar con PHP en el proceso de tecnología de web semántica.
- Se debe ampliar el presente trabajo a otro tipo de tecnologías de web semántica, que permitan completar la investigación y así crear más y mejores formas de ontologías.
- Se recomienda usar algoritmos de extracción y comparación de datos, para desarrollar los archivos de las ontologías.

## **GLOSARIO DE TERMINOS**

<b>ASCII</b>	American Standard Code for Information Interchange
<b>ESPOCH</b>	Escuela Superior Politécnica de Chimborazo
<b>HTML</b>	HyperText Markup Lenguaje
<b>LMS</b>	Learning Management System
<b>Moodle</b>	Modular Object-Oriented Dynamic Learning Environment
<b>NS</b>	Namespace
<b>OWL</b>	Ontology Web Language
<b>PHP</b>	Hypertext Pre-processor
<b>RDF</b>	Resource Description Framework
<b>SCORM</b>	Sharable Content Object Reference Model
<b>SGML</b>	Standard Generalized Markup Language
<b>SWRL</b>	Semantic Web Rule Language
<b>URI</b>	Uniform Resource Identifier
<b>VLEs</b>	Virtual Learning Environments
<b>XML</b>	eXtensible Markup Language

## BIBLIOGRAFÍA

**ANA TERESA GONZALEZ. 2009.** Guia de Apoyo para el uso de Moodle. [Online] Junio 29, 2009. [Cited: 11 20, 2014.] [https://download.moodle.org/docs/es/1.9.4\\_usuario\\_alumno.pdf](https://download.moodle.org/docs/es/1.9.4_usuario_alumno.pdf).

**ANTONIO GUERRERO, VÍCTOR A. VILLAGRÁ, JORGE E. LÓPEZ DE VERGARA. 2005.** <http://neweb.dit.upm.es/>. [Online] Septiembre 16, 2005. [Cited: Noviembre 03, 2013.] <http://neweb.dit.upm.es/~jlopez/publicaciones/jitel05-swrl.pdf>.

**CASTELLS P, R.BENJAMINS J.LORÉS, F. PERDRIX. 2011.** griho. [Online] febrero 21, 2011. [Cited: noviembre 16, 2013.] <http://griho2.udl.es/publicacions/2004/neptuno-interaccion04.pdf>.

**CASTELLS P, R.Benjamins J.Lorés, F. Perdrix. 2011.** <http://griho2.udl.es/>. [Online] febrero 21, 2011. [Cited: noviembre 6, 2013.] <http://griho2.udl.es/publicacions/2004/neptuno-interaccion04.pdf>.

**CASTELLS, FRANCISCO SAIZ Y PABLO. 2003.** <http://www.kramirez.net/>. [Online] octubre 16, 2003. [Cited: noviembre 6, 2013.] [http://www.kramirez.net/SMA\\_Maestria/Material/Presentaciones/2-ontologias.pdf](http://www.kramirez.net/SMA_Maestria/Material/Presentaciones/2-ontologias.pdf).

—. **2003.** [kramirez/](http://www.kramirez.net/). [Online] octubre 16, 2003. [Cited: noviembre 16, 2013.] [http://www.kramirez.net/SMA\\_Maestria/Material/Presentaciones/2-ontologias.pdf](http://www.kramirez.net/SMA_Maestria/Material/Presentaciones/2-ontologias.pdf).

**CASTELLS, PABLO. 2009.** <http://moodle2.unid.edu.mx/>. [Online] Mayo 28, 2009. [Cited: noviembre 12, 2013.] [http://moodle2.unid.edu.mx/dts\\_cursos\\_md/pos/TI/BE/AM/12/tecnicas\\_de\\_la\\_web\\_semantica.pdf](http://moodle2.unid.edu.mx/dts_cursos_md/pos/TI/BE/AM/12/tecnicas_de_la_web_semantica.pdf).

—. **2009.** [moodle2/](http://moodle2.unid.edu.mx/). [Online] Mayo 28, 2009. [Cited: noviembre 12, 2013.] [http://moodle2.unid.edu.mx/dts\\_cursos\\_md/pos/TI/BE/AM/12/tecnicas\\_de\\_la\\_web\\_semantica.pdf](http://moodle2.unid.edu.mx/dts_cursos_md/pos/TI/BE/AM/12/tecnicas_de_la_web_semantica.pdf).

**CEJUDO, MARIA DEL CARMEN LLORENTE. 2011.** De la Web tradicional a la Web semántica: cambios y aplicación al ámbito educativo. [Online] Abril 15, 2011. [Cited: Noviembre 12, 2013.] <http://www.udgvirtual.udg.mx/apertura/index.php/apertura3/article/view/191/206#per>.

**CODINA, LLUIS. 2003.** webs2002.uab.es/. <http://www.fti.uab.es/tradumatica/revista> . [Online] Noviembre 1, 2003. [Cited: Noviembre 6, 2013.] <http://webs2002.uab.es/tradumatica/revista/num2/articles/06/06.PDF>.

**CODINA, LLUÍS. 2003.** webs2002.uab.es/. <http://www.fti.uab.es/tradumatica/revista>. [Online] Noviembre 1, 2003. [Cited: Noviembre 16, 2013.] <http://webs2002.uab.es/tradumatica/revista/num2/articles/06/06.PDF>.

**CONSORTION, WORLD WIDE WEB. 2014.** Guías Breves de Tecnologías W3C. *Guías Breves de Tecnologías W3C*. [Online] enero 5, 2014. [Cited: septiembre 15, 2014.] Guías Breves de Tecnologías W3C.

**G.FALQUET. 2011.** <http://cui.unige.ch/>. [Online] mayo 5, 2011. [Cited: noviembre 15, 2013.] [http://cui.unige.ch/isi/icle-wiki/\\_media/cours:cours:sw:dl-rules-swrl.pdf](http://cui.unige.ch/isi/icle-wiki/_media/cours:cours:sw:dl-rules-swrl.pdf).

**GARCIA, RICARDO. 2008.** Estudio del impacto de las aplicaciones comerciales. [Online] 03 16, 2008. [Cited: 12 20, 2013.] <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/1861/1/42897.pdf>.

**GAVIOTA, IES. 2014.** El lenguaje HTML. *El lenguaje HTML*. [Online] enero 5, 2014. [Cited: septiembre 22, 2014.] <http://www.juntadeandalucia.es/averroes/iesgaviota/informatica/html.html>.

**H, FEHLERN. 2012.** planterdf.com. *planterdf.com*. [Online] febrero 19, 2012. [Cited: noviembre 5, 2013.] <http://planetrdf.com/>.

**H, FEHLERN. 2012.** planterdf.com. *planterdf.com*. [Online] febrero 19, 2012. [Cited: noviembre 15, 2013.] <http://planetrdf.com/>.

**HERNANDEZ, MARÍA FOLGUEIRA. 2014.** Master Universitario en eLearning y Redes Sociales. *Master Universitario en eLearning y Redes Sociales*. [Online] febrero 02, 2014. [Cited: septiembre 15, 2014.] <http://unirelearningmaster.blogspot.com/2014/02/moodle-dokeos-claroline-ventajas-e.html>.

**KSARMIENTO. 2009.** Semantizando la Web. [Online] noviembre 7, 2009. [Cited: noviembre 15, 2013.] <http://semantizandolaweb.wordpress.com/2011/11/07/que-es-rdf-y-para-que-es-bueno/>.

**LAGE, OSCAR. 2011.** SWRL. [Online] mayo 21, 2011. [Cited: noviembre 10, 2013.] <http://paginaspersonales.deusto.es/dipina/doctorado/presentaciones/SWRL.pdf>.

—. **2011.** SWRL. [Online] mayo 21, 2011. [Cited: noviembre 5, 2013.] <http://paginaspersonales.deusto.es/dipina/doctorado/presentaciones/SWRL.pdf>.

**LAPUENTE, MARÍA JESÚS LAMARCA. 2013.** La Web Semántica. [Online] 12 08, 2013. [Cited: Diciembre 10, 2013.] [http://www.hipertexto.info/documentos/web\\_semantica.htm](http://www.hipertexto.info/documentos/web_semantica.htm).

**LUFERN. 2010.** REICIS. [Online] Noviembre 2010. [Cited: Abril 01, 2015.] [www.ati.es/reicis](http://www.ati.es/reicis).

**MEDIAWIKI. 2014.** Wikipedia, enciclopedia libre. *Wikipedia, enciclopedia libre*. [Online] septiembre 20, 2014. [Cited: septiembre 22, 2014.] [http://es.wikipedia.org/wiki/Moodle#Caracter.C3.ADsticas\\_generales\\_de\\_Moodle](http://es.wikipedia.org/wiki/Moodle#Caracter.C3.ADsticas_generales_de_Moodle).

**MENDEZ, EVA. 2001.** SIDAR. *SIDAR traducciones*. [Online] enero 3, 2001. [Cited: Noviembre 24, 2015.] <http://www.sidar.org/recur/desdi/traduc/es/rdf/rdfesp.htm>.

**MUÑOZ, JOSE RIVERA. 2007.** *Tecnologías y Lenguajes para la Web Semántica RDF y Ontologías*. Madrid : Universidad Pontificia de Salamanca, 2007. 1.

**PABLO DÍEZ SUÁREZ, CARLOS TEJO ALONSO. 2008.** Grupo a. [Online] febrero 26, 2008. [Cited: Octubre 28, 2013.] <http://www.w3.org/2007/09/OWL-Overview-es.html>.

**PABLO DIEZ SUAREZ, Carlos Tejo Alonso. 2004.** W3C. [Online] febrero 10, 2004. [Cited: Noviembre 18, 2014.] [file:///C:/Users/Fabian/Downloads/Lenguaje%20de%20Ontolog%C3%ADas%20Web%20\\_\(OWL\\_%20del%20W3C\).pdf](file:///C:/Users/Fabian/Downloads/Lenguaje%20de%20Ontolog%C3%ADas%20Web%20_(OWL_%20del%20W3C).pdf).

**ROVIRA, LLUÍS CODINA Y CRISTÓFOL. 2006.** La web semantica1. [Online] Marzo 6, 2006. [Cited: Noviembre 12, 2013.] [www.semanticweb.net](http://www.semanticweb.net).

**RUIZ, RICARDO GARCÍA. 2008.** Estudio del impacto de las aplicaciones comerciales. [Online] 03 16, 2008. [Cited: Diciembre 10, 2013.] <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/1861/1/42897.pdf>.

**SEMANTICA, CAMBRIDGE. 2015.** Cambridge Semántica. *The Smart Data Advantage*. [Online] Julio 12, 2015. [Cited: Abril 15, 2016.] <http://www.cambridgesemantics.com/semantic-university/rdfs-vs-owl>.

**TELLO, ADOLFO LOZANO. 2010.** <http://eolo.cps.unizar.es>. [Online] mayo 5, 2010. [Cited: noviembre 26, 2013.] <http://eolo.cps.unizar.es/Docencia/MasterUPV/Articulos/No%20disponibles%20por%20estar%20seleccionados%20por%20otros/Ontologias%20en%20la%20Web%20Semantica.pdf>.

**TELLO, JESUS CACERES. 2013.** La Web Semántica y el lenguaje RDF. [Online] septiembre 10, 2013. [Cited: diciembre 11, 2015.] [https://portal.uah.es/portal/page/portal/GP\\_EPD/PG-MA-PROF/OLD\\_PG-PROF-138886%202008-07-14%2010-07-31/TAB4348465/TAB4348469/TAB4348477/Articulo\\_WebSemantica\\_Jesus\\_Caceres\\_CISTI\\_06.pdf](https://portal.uah.es/portal/page/portal/GP_EPD/PG-MA-PROF/OLD_PG-PROF-138886%202008-07-14%2010-07-31/TAB4348465/TAB4348469/TAB4348477/Articulo_WebSemantica_Jesus_Caceres_CISTI_06.pdf).

**W3C. 2004.** w3c.com. [Online] 02 10, 2004. [Cited: noviembre 14, 2013.] <http://www.w3.org/RDF/>.

## **ANEXOS**

### **Anexo A. MANUAL TÉCNICO**

#### **1. OBJETIVOS**

##### **1.1. Objetivo general**

Proporcionar una guía técnica a para los administradores del sistema en la cual se puedan apoyar para realizar labores de actualización o mantenimiento del sistema.

##### **1.2. Objetivos específicos**

- Mostrar la estructura física y lógica del sistema.
- Establecer la metodología utilizada para la elaboración del sistema.
- Describir las herramientas utilizadas para el desarrollo del sistema.

##### **1.3. Objetivo general del sistema**

Proveer a los técnicos y docentes de la Escuela Superior Politécnica de Chimborazo de una herramienta tecnológica que les permita obtener información de un curso virtual para de esta manera crear nuevos cursos a partir de los existentes.

#### **2. INFORMACIÓN GENERAL**

##### **2.1. Datos generales del proyecto**

**Nombre del proyecto:** Generador Ontologías



**Equipo de trabajo:** El desarrollo del presente proyecto está bajo la responsabilidad de Dalila Lucía Toledo Sanmartín estudiante de la Escuela de Ingeniería en Sistemas de la Facultad de Informática y Electrónica.

## **2.2. Datos generales de la institución**

**Nombre:**

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

**Dirección:**

Panamericana Sur km 1 1/2, Riobamba – Ecuador

## **2.3. Alcance**

La aplicación Generador Ontologías es una aplicación de web semántica, que interactúa con una base de datos de Moodle. Cuando se inicia la aplicación esta se conecta hacia la base de Moodle a partir de la cual se extrae la información de los cursos virtuales existentes.

## **3. CONTENIDO TÉCNICO**

### **3.1. Herramientas y equipos de desarrollo utilizadas**

#### **Hardware**

- Laptop HP Pavilion dv7

#### **Software**

- Entorno de desarrollo Netbeans
- Mysql
- ARC

## Requerimientos del sistema

- 100Mb de espacio en disco
- 1Gb de RAM
- Conexión de red activa
- Php 5.6 +

## 3.2. Modelo de desarrollo

### Modelo XP

Es una metodología ágil que se encarga de potenciar las relaciones interpersonales para el éxito de proyectos de desarrollo de software. Está enfocado al trabajo en equipo y se basa en la retroalimentación continua del cliente y el equipo de trabajo además de contar con una comunicación fluida y simplicidad en las soluciones implementadas. La programación extrema (XP) es una metodología de desarrollo ligera basada en una serie de valores y de prácticas de buenas maneras que persigue aumentar la productividad en el desarrollo de proyectos de software. El modelo XP básicamente busca dos objetivos.

1. Hacer un software con calidad
2. Hacerlo de la forma más rápida posible

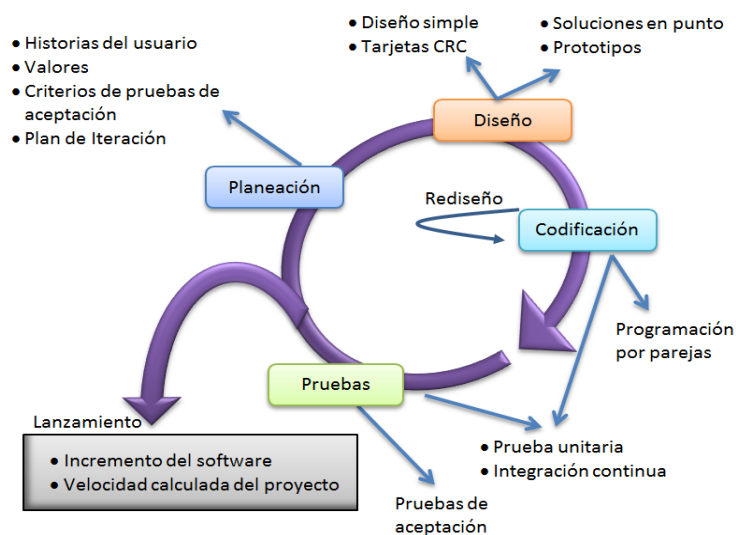


Figura 1-A. Modelo de Desarrollo XP

### **3.5.1. Fases de la metodología XP**

#### **Fase 1: Planificación del proyecto.**

En esta primera fase se debe hacer primero una recopilación de todos los requerimientos del proyecto, también debe haber una interacción con el usuario, y se debe planificar bien entre los desarrolladores del proyecto que es lo que se quiere para el proyecto para así lograr los objetivos finales.

#### **Fase 2: Diseño.**

La metodología utilizada pretende la creación de diseños simples y sencillos. Para procurar hacerlo todo lo menos complicado posible para el usuario o cliente, para conseguir un diseño fácilmente entendible por el usuario que a la larga facilitara el desarrollo, ahorrando tiempo y esfuerzo. En esta fase se consigue crear la parte física la interfaz que tendrá el usuario o cliente con el proyecto.

#### **Fase 3: Codificación.**

Para un mejor desarrollo se involucra al cliente dentro del proceso de desarrollo ya que su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Al momento de iniciar el desarrollo de cada historia de usuario el cliente especifica su requerimiento de manera detallada para de esta manera el requerimiento sea cumplido y el cliente pueda estar presente durante las pruebas funcionales para que al final del desarrollo el sistema quede apto y acorde con las necesidades del usuario. En esta fase está incluido todo lo de codificación o programación por parte de los desarrolladores del proyecto.

#### Fase 4: Pruebas.

Esta fase es una de las más relevantes de la metodología XP ya que los test se realizan a medida que se avanza con el desarrollo, de esta manera se puede asegurar que el sistema cumpla con las expectativas planteadas al inicio. Es decir en esta fase se van realizando test (pruebas) al código implementado por cada requerimiento.

#### 3.6. Arquitectura de software

La arquitectura utilizada para el desarrollo de la aplicación Generador Ontologías es la MVC (Modelo Vista Controlador), la misma que permite codificar de manera óptima cada uno de los recursos de la aplicación.

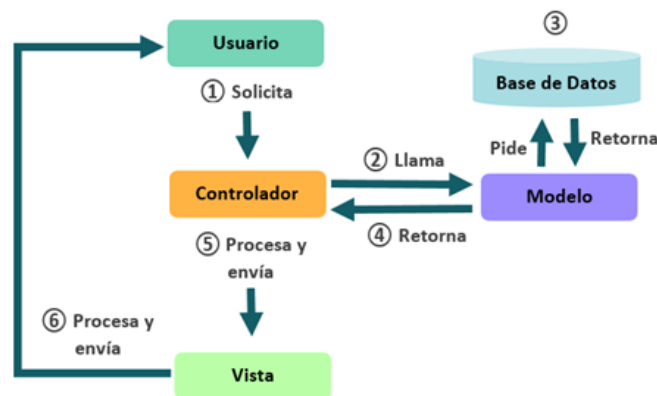


Figura 6-A. Modelo Vista Controlador.

**Fuente:** <https://www.google.com.ec/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=&url=http%3A%2F%2Frodrigr.com%2Fblog%2Fpage%2F%2F&psig=AFQjCNEv e4ORy6wkg3RYVxJGyJjwfgizw&ust=1473985040822810>

#### 3.7. Restricciones

##### Del sistema

50 Mb de espacio en disco

1Gb de RAM

PHP 5.6 +

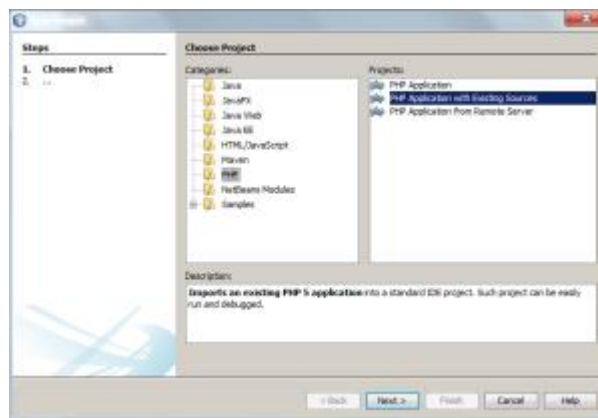
### 3.8. Modelo de datos

#### 3.8.1. Base de Datos

La base de datos a la cual se conecta la aplicación es la base de Moodle, a partir de la cual se realizan las consultas para extraer los datos que se requieren para el instanciamiento de la ontología.

#### 3.8.2. Creación de un Proyecto en NetBeans

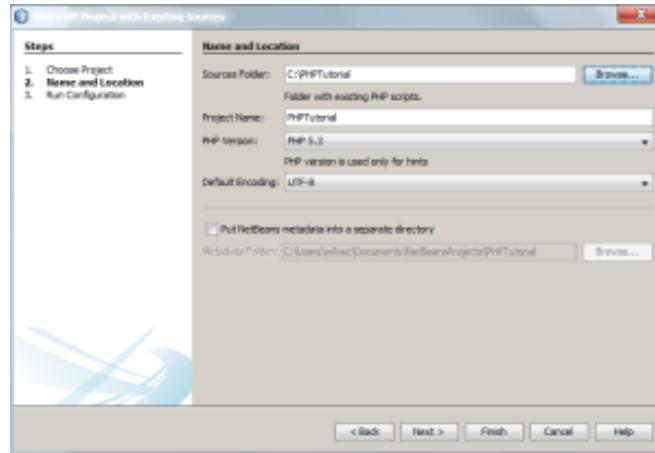
Abrimos Netbeans y Hacemos clic en File > New Project. En *Categories* seleccionamos PHP y en *Project* seleccionamos *PHP Application with Existing Sources*. Hacemos click en Next



**Figura 7-A.** Nuevo Proyecto PHP a partir de base existente

**Realizado por:** Dalila Toledo

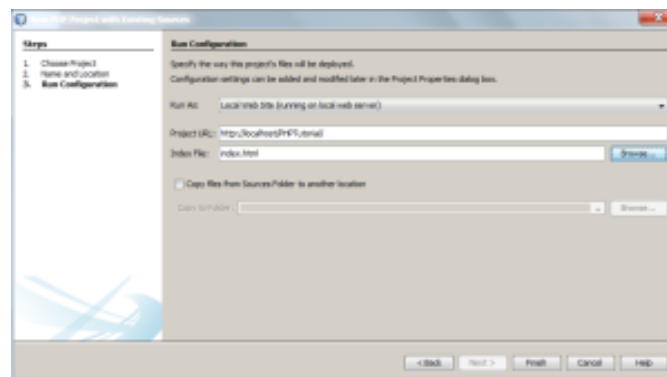
En la parte *Name and Location*, en *Source Folder*, buscamos la ruta de nuestra carpeta. Los demás valores del cuadro de diálogo los dejamos como están y hacemos click en *Next*



**Figura 8-A.** Selección archivo

**Realizado por:** Dalila Toledo

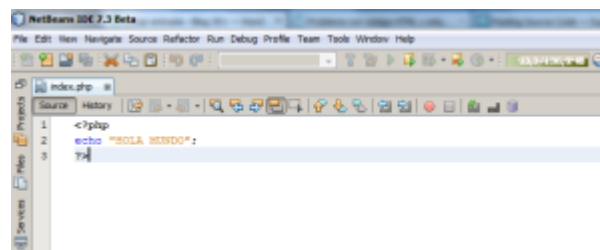
En la parte Run configuration, en el campo Project URL colocamos la URL de nuestro proyecto (en mi caso: <http://localhost/PHPTutorial>). en el campo Index File, escogemos el archivo index.php que está en la carpeta.



**Figura 9-A.** Comprobación de la ruta

**Realizado por:** Dalila Toledo

Podemos ver que inmediatamente nos abre el archivo por defecto.



**Figura 10-A.** Se crea el proyecto

**Realizado por:** Dalila Toledo

### 3.8.3. Codificación

La arquitectura utilizada para el desarrollo de la aplicación es la MVC (Modelo Vista Controlador) ya esta arquitectura es una de las mejores en cuanto a escalabilidad. Para la obtención de los datos de un curso a partir de la base de Moodle, se utilizan consultas de sql como:

#### **LISTAR LOS CURSOS EXISTENTES**

```
SELECT id, fullname FROM `mdl_course`
```

#### **LISTAR LAS SECCIONES DE UN CURSO**

```
SELECT * FROM `mdl_course_sections` WHERE `course`=$variable and `summary`<>""
```

#### **CONTAR LAS SECCIONES DE UN CURSO**

```
SELECT count(`id`) FROM `mdl_course_sections` WHERE `course`=$variable and `summary`<>""
```

#### **LISTAR LOS BLOQUES DE LA DERECHA**

```
SELECT i.`id`,i.`blockname`
```

```
FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` = p.`blockinstanceid`
```

```
WHERE p.region='side-post'
```

#### **LISTAR LOS BLOQUES DE LA IZQUIERDA**

```
SELECT i.`id`,i.`blockname`
```

```
FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` = p.`blockinstanceid`
```

```
WHERE p.region='side-pre'
```

#### **CONTAR LOS BLOQUES DE LA DERECHA**

```
SELECT COUNT( i.`id`)
```

```
FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` = p.`blockinstanceid`
```

```
WHERE p.region='side-post'WHERE p.region='side-post'
```

#### **CONTAR LOS BLOQUES DE LA IZQUIERDA**

```

SELECT COUNT( i.`id`)

FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` =
p.`blockinstanceid`

WHERE p.region='side-pre'

```

### **LISTAR LOS EXAMENES QUE TIENE UN CURSO**

```

SELECT count(`id`) FROM `mdl_quiz` WHERE `course`=$variable

```

### **CONTAR LOS EXAMENES QUE TIENE UN CURSO**

```

SELECT count(`id`) FROM `mdl_quiz` WHERE `course`=$variable

```

### **LISTAR LOS FOROS DE UN CURSO**

```

SELECT `id`,`name` FROM `mdl_forum` WHERE `course`=$variable

```

### **CUANTOS FOROS TIENE UN CURSO**

```

SELECT count(`id`) FROM `mdl_forum` WHERE `course`=$variable

```

### **LISTAR LOS RECURSOS DE UN CURSO**

```

SELECT `id`,`name` FROM `mdl_resource` WHERE `course` = $variable

```

### **CUANTOS RECURSOS TIENE UN CURSO**

```

SELECT count(`id`) FROM `mdl_resource` WHERE `course`=$variable

```

Para la creacion del archivo `.owl` se trabaja mediante funciones las mismas que están basadas en las consultas realizadas anteriormente.

La función para general el archivo `.owl` es:

```

public static function getOntologia($details) {
    $ind = "";
    foreach (($details->details) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'. $element->name. '-->'
        . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#'.
        $element->name. '">'
    }
}

```



```

    .<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
    >tipo ."/>'
    .</owl:NamedIndividual>;
    $ind .= $tripleta;
} foreach (($details->details1) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
    .<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
    $element->name ."'>'
    .<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
    >tipo ."/>'
    .</owl:NamedIndividual>;
    $ind .= $tripleta;
} foreach (($details->details2) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
    .<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
    $element->name ."'>'
    .<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
    >tipo ."/>'
    .</owl:NamedIndividual>;
    $ind .= $tripleta;
} foreach (($details->details3) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
    .<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
    $element->name ."'>'
    .<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
    >tipo ."/>'
    .</owl:NamedIndividual>;
    $ind .= $tripleta;
} foreach (($details->details4) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'. $element->name. '-->'
    .<owl:NamedIndividual rdf:about="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#'.
    $element->name ."'>'
    .<rdf:type rdf:resource="http://webapp.esepoch.edu.ec/ontolms/cursol.owl#". $element-
    >tipo ."/>'

```

```

        .'/</owl:NamedIndividual>';
        $ind .= $tripleta;
    } foreach (($details->details5) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
        .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
        $element->name ."'>'
        .'/<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
        >tipo ."'/>'
        .'/</owl:NamedIndividual>';
        $ind .= $tripleta;
    } foreach (($details->details6) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
        .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
        $element->name ."'>'
        '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
        >tipo ."'/>'
        .'/</owl:NamedIndividual>';
        $ind .= $tripleta;
    } foreach (($details->details7) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
        .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
        $element->name ."'>'
        .'/<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
        >tipo ."'/>'
        .'/</owl:NamedIndividual>';
        $ind .= $tripleta;
    } foreach (($details->details8) as $element) {
        $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
        .'/<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
        $element->name ."'>'
        .'/<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
        >tipo ."'/>'
        .'/</owl:NamedIndividual>';
        $ind .= $tripleta;
    }

```

```

} foreach (($details->details9) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    . '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
    >tipo ."'>'
    . '</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details10) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    . '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
    >tipo ."'>'
    . '</owl:NamedIndividual>';
    $ind .= $tripleta;
} foreach (($details->details11) as $element) {
    $tripleta = '<!--http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element->name. '-->'
    . '<owl:NamedIndividual    rdf:about="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#.
    $element->name ."'>'
    . '<rdf:type    rdf:resource="http://webapp.esepoch.edu.ec/ontolms/curso1.owl#. $element-
    >tipo ."'>'
    . '</owl:NamedIndividual>';
    $ind .= $tripleta;
}

//Obtiene cabecera de la ontologia
$pathfile = app_path()."/files/ontologia.owl";
$content = file_get_contents($pathfile);
$content = str_replace("%individuals%", $ind, $content);

//Crea y guarda el contenido de la ontologia en curso.owl
$pathfile = app_path()."/files/curso.owl";
if (is_file($pathfile)) {
    unlink($pathfile);
}

```

```

}
$handle = fopen($pathfile, "w+");
fwrite ($handle,$content);
fclose($handle);
}

```

La función para obtener los recursos y toda la información de un curso es:

```

public static function getCourse($id) {
    $sql = "SELECT id, fullname FROM mdl_course WHERE id = ". $id;
    $header = DB::select(TRIM($sql));
    $query = "SELECT id, name, 'Recurso' as tipo FROM mdl_resource WHERE course = ". $id;
    $detail = DB::select(TRIM($query));
    $query1 = "SELECT `id`, `name`, 'Tarea' as tipo FROM `mdl_assignment` WHERE `course`
= ". $id;
    $detail1 = DB::select(TRIM($query1));
    $query2 = "SELECT i.`id`,i.`blockname` as name, 'Bloque' as tipo
        FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` =
p.`blockinstanceid`
        WHERE p.region='side-post'
        Group by i.`blockname`";
    $detail2 = DB::select(TRIM($query2));
    $query3 = "SELECT i.`id`,i.`blockname` as name, 'Bloque' as tipo
        FROM `mdl_block_instances` i INNER JOIN `mdl_block_positions` p ON i.`id` =
p.`blockinstanceid`
        WHERE p.region='side-pre'
        Group by i.`blockname`";
    $detail3 = DB::select(TRIM($query3));
    $query4 = "SELECT `id`, `name`, 'Libro' as tipo FROM `mdl_book` WHERE `course` = ".
$id;
    $detail4 = DB::select(TRIM($query4));
    $query5 = "SELECT id, section, 'Seccion' as tipo FROM `mdl_course_sections` WHERE
`course` = ". $id. " and `summary`<>''";
    $detail5 = DB::select(TRIM($query5));

```

```

$query6 = "SELECT `id`, `name`, 'Foro' as tipo FROM `mdl_forum` WHERE `course` = ".
$Id;
$detail6 = DB::select(TRIM($query6));
$query7 = "SELECT `id`, `name`, 'Etiqueta' as tipo FROM `mdl_label` WHERE `course` = ".
$Id;
$detail7 = DB::select(TRIM($query7));
$query8 = "SELECT `id`, `name`, 'Pagina' as tipo FROM `mdl_page` WHERE `course` = ".
$Id;
$detail8 = DB::select(TRIM($query8));
$query9 = "SELECT `id`, `name`, 'Prueba' as tipo FROM `mdl_quiz` WHERE `course` = ".
$Id;
$detail9 = DB::select(TRIM($query9));
$query10 = "SELECT `id`, `name`, 'Url' as tipo FROM `mdl_url` WHERE `course` = ". $Id;
$detail10 = DB::select(TRIM($query10));
$query11 = "SELECT `id`, `name`, 'Wiki' as tipo FROM `mdl_wiki` WHERE `course` = ". $Id;
$detail11 = DB::select(TRIM($query11));
$result = [
    'title' => $header[0],
    'details' => $detail,
    'details1' => $detail1,
    'details2' => $detail2,
    'details3' => $detail3,
    'details4' => $detail4,
    'details5' => $detail5,
    'details6' => $detail6,
    'details7' => $detail7,
    'details8' => $detail8,
    'details9' => $detail9,
    'details10' => $detail10,
    'details11' => $detail11    ];
$curso = (object)$result;
return $curso;
}

```

## **Anexo B: MANUAL DE USUARIO**

### **PROPÓSITO**

Este manual describe detalladamente las instrucciones del manejo de la aplicación Generador Ontologías. Se recomienda la lectura de este manual antes de usar la aplicación.

### **ALCANCE**

Este manual está orientado a usuarios de la aplicación Generador Ontologías. Pretende servir de referencia y de introducción a la aplicación.

## **1. OBJETIVO**

### ***1.1. Objetivo General***

Este documento tiene como objetivo principal servir como guía de uso del sistema para los usuarios directos como son: Técnicos y docentes de la ESPOCH.

### **1.2. Objetivos Específicos**

- Enseñar el manejo y funcionamiento de cada una de las funciones de la aplicación.
- Dar guías para la solución de algunos problemas o complicaciones que se tenga al utilizar la aplicación.

## **2. INFORMACIÓN GENERAL**

### **2.1. Datos generales del proyecto**

**Nombre del proyecto:** Generador Ontologías

**Equipo de trabajo:** El desarrollo del presente proyecto está bajo la responsabilidad de la estudiante de la Escuela de Ingeniería en Sistemas de la Facultad de Informática y Electrónica:

Dalila Lucia Toledo Sanmartín

## **2.2. Datos generales de la institución**

**Nombre:**

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

**Dirección:**

Panamericana Sur km 1 1/2, Riobamba – Ecuador

## **2.3. Descripción de la aplicación**

La aplicación Generador Ontologías es una aplicación de web semántica, que interactúa con una base de datos de Moodle. Cuando se inicia la aplicación esta se conecta hacia la base de Moodle a partir de la cual se extrae la información de los cursos virtuales existentes.

### **Requerimientos de sistema**

- 100Mb de espacio en disco
- 1Gb de RAM
- Conexión de red activa
- Php 5.6 +

## 2.4. Funcionamiento de la Aplicación

- a. **Inicio de la aplicación:** Esta aplicación puede ser ejecutada por técnicos y maestros de la ESPOCH mediante su conexión de la siguiente manera:

En el url del navegador escriba: *webapp.esPOCH.edu.ec/ontologías/public* y se abrirá la aplicación como se observa en la figura 7-B.

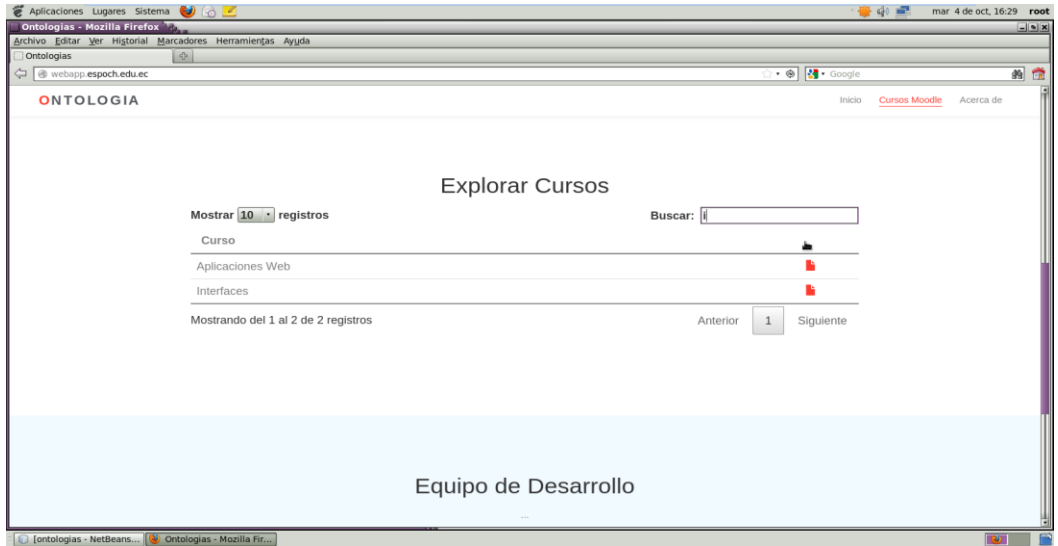


**Figura 1-B. Inicio de Aplicación**

**Realizado por:** Dalila Toledo

- b. **Búsqueda de información:** El usuario deberá ubicarse en la parte de *Cursos Moodle*, en donde ingresara el nombre del curso que desea buscar para que posteriormente la aplicación extraiga los cursos cuyos nombres tengan coincidencias con lo solicitado por el usuario.

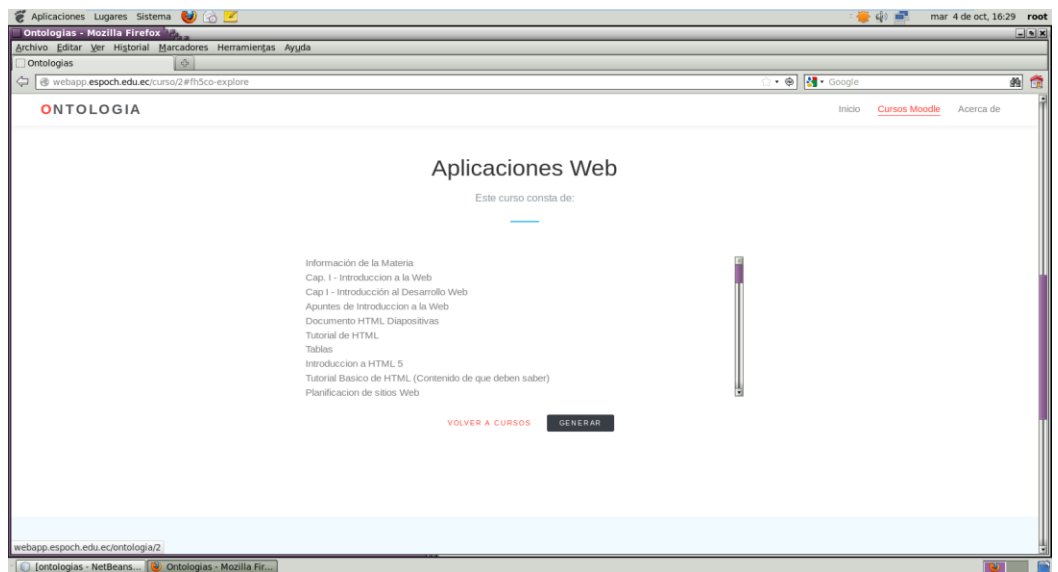




**Figura 2-B. Búsqueda de Cursos**

**Realizado por:** Dalila Toledo

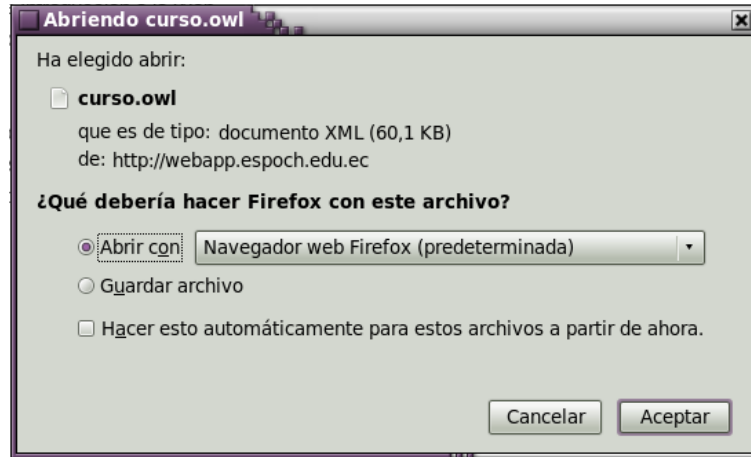
- c. **Presentación de resultados:** Una vez realizada la búsqueda, la aplicación presenta un listado de las coincidencias, a partir de las cuales el usuario puede elegir un curso para ver sus contenidos, dando clic en el icono de la derecha. Posteriormente se abrirá una pantalla con la información del curso seleccionado como se observa en la figura 9-B.



**Figura 3-B. Presentación de Resultados**

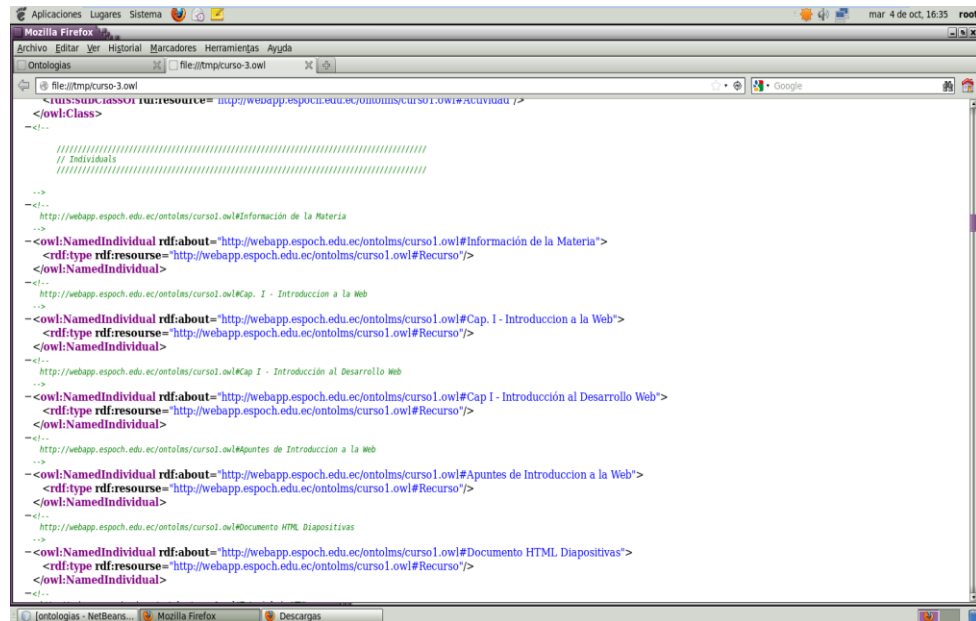
**Realizado por:** Dalila Toledo

- d. **Generar:** Al dar clic en el botón generar se precede a crear el archivo .owl en el cual se encuentra estructurada la ontología con los recursos obtenidos de un curso. El archivo .owl podrá ser visualizado en un navegador o a su vez también se podrá descargar en un archivo independiente.



**Figura 4-B.** Generación del Archivo .owl

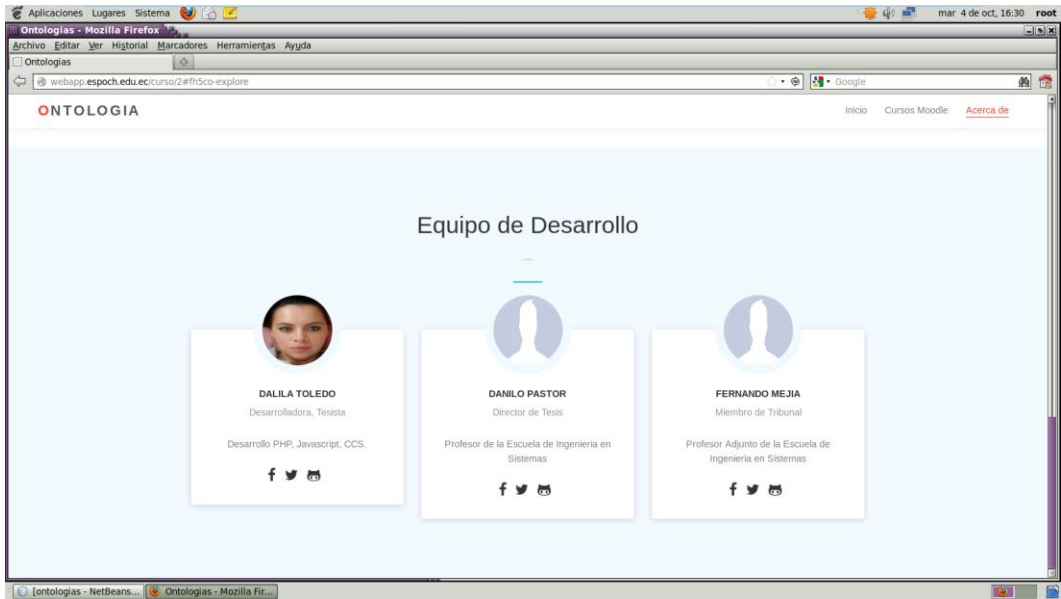
Realizado por: Dalila Toledo



**Figura 5-B.** Archivo owl generado

Realizado por: Dalila Toledo

- e. **Acerca de:** En la opción de acerca de se encuentra la información del equipo de trabajo que estuvo a cargo del desarrollo de la aplicación.



**Figura 6-B.** Información del equipo de desarrollo

**Realizado por:** Dalila Toledo