

Universidad de las Ciencias Informáticas

Facultad 6



Centro de Geoinformática y Señales Digitales

**Enriquecimiento de una Ontología Geográfica,
por medios semiautomáticos, a partir de fuentes
de información disponibles en Internet**


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

Autor: Yerandy Hernández Barreras

Tutor: Msc. Manuel Enrique Puebla Martínez

Ciudad de la Habana, julio 2015

“Año 57 de la Revolución”



*"Si no puedes volar, corre.
Si no puedes correr, camina.
Si no puedes caminar, gatea.
Pero hagas lo que hagas,
siempre sigue hacia adelante"*

Martin Luther King

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este Trabajo de Diploma titulado Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes de información disponibles en Internet y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría en La Habana a los 1 días del mes de julio del año 2015.

Yerandy Hernández Barreras

Autor

Msc. Manuel Enrique Puebla Martínez

Tutor

DATOS DEL CONTACTO

Tutor: Msc. Manuel Enrique Puebla Martínez.

Sexo: M.

Institución: Universidad de las Ciencias Informáticas (UCI).

Dirección de la Institución: Carretera San Antonio de los Baños, Km 2 $\frac{1}{2}$ Boyeros.

Correo electrónico: mpuebla@uci.cu.

Teléfono del apto: 7835-87 53.

Síntesis del Tutor: Licenciado en ciencia de la computación de la Universidad de Oriente en el año 2004. Profesor asistente del departamento de programación de la facultad 6 de la Universidad de las Ciencias Informáticas. En el año 2008 defendió su tesis de maestría en Nuevas Tecnologías de la Educación. Desde el 2010 está vinculado a la línea de investigación Semántica Espacial. Pertenece al grupo de investigación de web semántica de la UCI (<http://gws-uci.blogspot.com/>).

AGRADECIMIENTOS

A MIS DOS MADRES LIDIA MARTA Y FORTUNATA POR DARME SU AMOR, SU APOYO Y EDUCARME DESDE PEQUEÑO, GUIARME Y APOYARME INCONDICIONALMENTE.

A MIS TÍAS MELEN, JULITA, IRIS Y MI TÍO JORGE POR SU CARIÑO Y PREOCUPACIÓN DESDE NIÑO.

A MIS PRIMOS ALEXIS, DENNYS Y LUIS ENRIQUE, ADEMÁS DE PRIMOS, HERMANOS, AMIGOS Y COMPAÑEROS.

A JUANITA.

A MI PRIMA DAYLÍN Y MI PRIMITO CELSO YUDÁN.

A IRACEMA Y PEPE, MIS SEGUNDOS PADRES.

A JANET, JORGITO Y CHÉSTER POR SU PREOCUPACIÓN Y ATENCIÓN.

A MIS COMPAÑEROS Y COMPAÑERAS DE ESCUELA, CON LOS QUE HE VIVIDO BUENOS Y MALOS MOMENTOS DE MI VIDA.

A MIS COMPAÑEROS DEL BARRIO, QUE AUNQUE ESTÁN LEJOS SIEMPRE ME APOYABAN Y ME DABAN ALIENTO PARA CONTINUAR.

A LOS PROFESORES ROSAYDA VALIENTE, YURIMA IBAÑES, NARA LIDIA, YUDELKIS ABAD, JORGE EMILIO (JEEM), JANET CRISTINA QUE TRANSMITIERON SU CONOCIMIENTO HACIA MÍ DESDE LOS INICIOS EN LA UNIVERSIDAD Y ESTUVIERON PRESENTES CUANDO LO NECESITÉ.

A LA PROFE LISBET, POR SU ATENCIÓN CADA VEZ QUE SURGÍA UNA DUDA.

A ELEYDIS, POR SU CONFIANZA EN MÍ DESDE EL PRINCIPIO.

A MI TUTOR, POR SU PACIENCIA Y ATENCIÓN CADA VEZ QUE NECESITAMOS SU AYUDA.

A TODAS LAS AQUELLAS PERSONAS QUE DE UNA FORMA U OTRA CONTRIBUYERON A QUE ESTE SUEÑO SE REALIZARA.

GRACIAS.

DEDICATORIA

LE DEDICO MI ESFUERZO, RESULTADO DE ESTE TRABAJO A:

MIS DOS MADRES LIDIA MARTA Y FORTUNATA, POR SER MADRES Y PADRES AL MISMO TIEMPO. LAS QUIERO MUCHO. SIN USTEDES NUNCA HUBIESE HECHO REALIDAD MI SUEÑO DE CONVERTIRME EN UN INGENIERO INFORMÁTICO.

MI TÍO PIOBES, TÍO Y PADRE, QUE AUNQUE NO SE ENCUENTRE CONMIGO, SÉ QUE ESTÁ ORGULLOSO DE MI RESULTADO. QUE DIOS LO TENGA EN LA GLORIA.

TODOS MIS FAMILIARES, QUE HAN ESTADO AL TANTO DE MIS ESTUDIOS DURANTE 24 AÑOS.

RESUMEN

A través de los años, con el creciente desarrollo de internet, el hombre ha intentado tener a su alcance la mayor cantidad de información posible; accesible y con la mayor riqueza que puedan poseer los datos. De aquí surgen las ontologías, con el propósito de brindar información semántica sobre los datos. Las ontologías proveen definiciones formales explícitas de entidades temáticas y sus relaciones, por lo tanto, facilitan las definiciones de métodos para proyectar, trasladar e integrar información obtenida de diferentes fuentes. Con la presente investigación se pretende automatizar parcialmente el enriquecimiento de una ontología geográfica, a partir de fuentes disponibles en internet. Esto aporta como principal ventaja, disminuir el costo en tiempo y esfuerzo en el desarrollo y mantenimiento de ontologías geográficas de gran tamaño, eliminando errores cometidos que pueden introducirse de forma manual. Se lleva a cabo el cumplimiento de este objetivo, realizando un análisis de los elementos teóricos necesarios para lograr la comprensión del tema ontológico; la creación, el consumo y la publicación de servicios web. Además, se utiliza en la construcción del sistema herramientas como el NetBeans, Visual Paradigm y la plataforma soberana para desarrollo web GENESIG. Con el desarrollo de la herramienta de software vinculada a la plataforma, se logra enriquecer una ontología del dominio geográfico, brindando información semántica sobre los datos geográficos.

Palabras claves: enriquecimiento semiautomático, internet, ontología geográfica, semántica espacial, servicios web.

ABSTRACT

ABSTRACT

Through the years, with the increasing development of internet, the man has tried to have at their fingertips as much information as possible; accessible and with the greatest veracity the data may have. From here ontologies arise, in order to provide semantic information about the data. Ontologies provide explicit formal definition of thematic entities and their relationships, therefore, they facilitate the definition of methods for projecting, transferring and integrating information from different sources. With the present research it is intended partially automate the enrichment of a geographic ontology from sources available online. This provides as main advantage, to reduce the cost in time and effort in the development and maintenance of large geographic ontologies, eliminating errors made that can be entered manually. To fulfill this objective, it is made and analysis the necessary theoretical elements to achieve understanding of the ontological subject: the creation, consumption and publication of Web Services. In addition, it is used in the creation of the system tools such as NetBeans, Visual Paradigm and GENESIG sovereign platform for web development. With the development of the software tool linked to the platform, it is possible to enrich an ontology of geographic domain, providing semantic information about geographic data.

Keywords: semiautomatic enrichment, internet, geographic ontology, semantic information, web services.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Marco Teórico	6
1.2 Fuentes de información	12
1.2.1 Centro de Excelencia para los Servicios de Información Geográfica.....	12
1.2.2 Wikipedia	13
1.2.3 GeoNames.....	13
1.2.4 Análisis de las fuentes de información	14
1.3 Estado del Arte.....	15
1.4 Herramientas para gestionar ontologías	17
1.4.1 Chimaera	18
1.4.2 Ontolingua	18
1.4.3 Protégé	18
1.5 Metodología de desarrollo de software	19
1.5.1 Metodología tradicional	19
1.5.2 Metodología ágil.....	20
1.5.3 Agile Unified Process (AUP)	20
1.6 Herramientas y tecnologías	22
1.6.1 Visual Paradigm for UML 8.0	22
1.6.2 Lenguaje Unificado de Modelado 8.0.....	23
1.6.3 Lenguaje de programación del lado del servidor	23
1.6.4 Lenguaje de programación del lado del cliente	25
1.6.5 IDE NetBeans 8.0	25
1.6.6 WSDL	26
1.6.7 Servidor Web Glassfish 4.0.....	26
1.6.8 Apache Jena 2.12.0	26
1.6.9 GeoNames-Source 1.1.12.....	27
1.7 GENESIG.....	27
1.7.1 Herramientas y tecnologías de la plataforma GENESIG	27
1.7.2 Servidor Web Apache 2.2	28
1.7.3 Arquitectura de GENESIG.....	28
CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN	29

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

2.1 Propuesta de solución	29
2.2 Modelo de dominio	30
2.3 Especificación de requisitos	31
2.3.1. Requisitos funcionales	31
2.3.2. Requisitos no funcionales	32
2.4 Descripción de la solución propuesta	33
2.4.1 Diagrama de Caso de Uso	33
2.4.2 Priorización de casos de uso	34
2.4.3 Especificación de los casos de uso	34
2.5 Arquitectura de software	46
2.5.1 Arquitectura orientada a objetos	47
2.5.2 Arquitectura basada en componentes	47
2.6 Patrones de diseño	47
2.6.1 Patrones GRASP	47
2.6.2 Patrones GoF	48
2.7 Modelo de diseño	49
2.7.1 Diagrama de clases del diseño	49
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS	53
3.1 Modelo de implementación	53
3.1.1 Diagrama de componentes	53
3.1.2 Diagrama de Despliegue	54
3.2 Pruebas del sistema propuesto	55
3.2.1 Pruebas de caja negra	55
CONCLUSIONES	58
RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS	60
ANEXOS	63

Índice de tablas

<i>Tabla 1: Actores del Sistema.</i>	34
<i>Tabla 2: Descripción del Caso de uso Enriquecer ontología.</i>	34
<i>Tabla 3: Descripción del Caso de uso Gestionar conceptos.</i>	39
<i>Tabla 4: Descripción del Caso de uso Gestionar instancias.</i>	41
<i>Tabla 5: Tabla 4: Descripción del Caso de uso Gestionar propiedades.</i>	43
<i>Tabla 6: Descripción del Caso de uso Guardar ontología.</i>	45
<i>Tabla 7: Diseño de Casos de Prueba para el caso de uso Enriquecer ontología.</i>	56
<i>Tabla 8: Diseño de Casos de Prueba para el caso de uso Guardar ontología.</i>	65
<i>Tabla 9: Diseño de Casos de Prueba para el caso de uso Gestionar conceptos.</i>	65
<i>Tabla 10: Diseño de Casos de Prueba para el caso de uso Gestionar instancias.</i>	65
<i>Tabla 11: Diseño de Casos de Prueba para el caso de uso Gestionar propiedades.</i>	66

Índice de figuras

<i>Figura 1: Tipos de ontologías.....</i>	<i>7</i>
<i>Figura 2: Disciplinas, fases e iteraciones de AUP.....</i>	<i>21</i>
<i>Figura 3: Modelo de dominio.....</i>	<i>31</i>
<i>Figura 4: Diagrama de Casos de uso.</i>	<i>33</i>
<i>Figura 5: Diagrama de clases del diseño del Caso de Uso Enriquecer ontología.....</i>	<i>51</i>
<i>Figura 6: Diagrama de componentes del Caso de Uso Enriquecer ontología.....</i>	<i>54</i>
<i>Figura 7: Diagrama de despliegue.....</i>	<i>55</i>
<i>Figura 8: Resultados de las pruebas aplicadas.</i>	<i>57</i>
<i>Figura 9: Diagrama de clases del diseño del Caso de Uso Gestionar conceptos.</i>	<i>63</i>
<i>Figura 10: Diagrama de clases del diseño del Caso de Uso Gestionar instancias.</i>	<i>63</i>
<i>Figura 11: Diagrama de clases del diseño del Caso de Uso Gestionar propiedades.....</i>	<i>64</i>
<i>Figura 12: Diagrama de clases del diseño del Caso de Uso Guardar ontología.</i>	<i>64</i>

INTRODUCCIÓN

INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC), son un conjunto de tecnologías desarrolladas para encargarse de la gestión de la información a través de diferentes herramientas que permiten el ahorro de los recursos humanos, materiales y la disminución del tiempo de trabajo. Comprenden una gran cantidad de soluciones e incluyen tecnologías para el almacenamiento de la información, su recuperación, su envío y recibo o procesamiento.

Históricamente el hombre ha intentado almacenar información, ya sea de forma manual o utilizando la tecnología. Esto ha requerido la búsqueda de nuevas herramientas, que permitan almacenar de manera más rápida, mayor potencial de información, permitiendo así optimizar el trabajo. Una de las técnicas utilizadas para solucionar este problema ha sido el almacenamiento de la información en la web, representándola y añadiendo semántica sobre la misma.

La semántica, según la (RAE, 2015), es lo perteneciente o relativo al significado de las palabras. La Web Semántica es una web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla, gracias a una información mejor definida. Al dotar a la web de más significado, se pueden obtener soluciones a problemas habituales en la búsqueda de información, gracias a la utilización de una infraestructura común, mediante la cual es posible compartir, procesar y transferir información de forma sencilla. Gracias a la semántica en la web, el software es capaz de procesar su contenido, razonar con este, combinarlo y realizar deducciones lógicas para resolver problemas cotidianos automáticamente (W3C, 2015).

Desde la aparición de la Inteligencia Artificial, las ontologías juegan un papel de gran importancia, ya que otorgan información a los sistemas de conocimientos. El objetivo es que los sistemas “razonen” de manera automática en dependencia de la información implícita que poseen y logren tomar decisiones.

Con la creación de las ontologías, se logra añadir semántica sobre datos. Las ontologías proveen definiciones formales explícitas de entidades temáticas y sus relaciones, por lo tanto, facilitan las definiciones de métodos para proyectar, trasladar e integrar información obtenida de diferentes fuentes.

Varios autores defienden las ontologías como un medio de entendimiento entre las personas y los sistemas informáticos. Según (Gruber, 1993), las ontologías tienen un papel clave para estructurar, manejar, compartir y procesar la información en la Web Semántica, comercio electrónico y aplicaciones médicas; ya que proveen conocimiento compartido y común de un determinado dominio. El conocimiento compartido permite a las personas y aplicaciones de cómputo comunicarse de manera efectiva. Mientras que (Garea Llano, 2007) explica que la razón de la popularidad de las ontologías es que ofrecen un

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

INTRODUCCIÓN

entendimiento común y compartido de algunos dominios que pueden ser comunicados entre las personas y los sistemas aplicados.

En la Línea de Productos de Software (LPS) Aplicativos SIG, del centro productivo Geoinformática y Señales Digitales (GEYSED), que pertenece a la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI), se desarrolla una plataforma tecnológica denominada GENESIG, destinada a agilizar el desarrollo de los Sistemas de Información Geográfica (SIG) en la web, reutilizando sus módulos y funcionalidades. Una de las metas a alcanzar por parte del proyecto es lograr una fortaleza tecnológica del software libre, añadiendo funcionalidades que aumenten el valor científico de la plataforma; sobre todo que permitan disminuir cada día más el tiempo de desarrollo de los SIG que se elaboran con ella (Martínez, y otros, 2012).

Como parte de un primer alcance, se ha decidido priorizar la gestión de topónimos¹ dentro de la plataforma GENESIG, por el valor de uso que tiene para los usuarios de los SIG, y además, desarrollar un módulo de gestión de topónimos que sea capaz de utilizar una ontología geográfica o geontología como forma de representación del conocimiento asociado a los lugares y posteriormente alimentarse de dicho conocimiento para garantizar las operaciones de búsqueda (Martínez, y otros, 2012).

El enriquecimiento manual de una ontología es una labor enormemente compleja y costosa en recursos humanos, técnicos, y de tiempo. El problema principal radica en que la persona debe poseer un vasto conocimiento sobre los componentes de una ontología y sus características, para lograr añadir los nuevos elementos a la ontología deseada. También, debe tener al menos una idea de cómo trabajar con algunas de las herramientas de edición, mantenimiento o creación de ontologías.

Tanto el primer como el segundo aspecto son difíciles de dominar, pues se debe realizar un profundo estudio sobre el tema ontológico para alcanzar un buen dominio y así analizar cuál de las herramientas presenta una menor curva de aprendizaje, y que al mismo tiempo cumpla satisfactoriamente el objetivo trazado.

Los seres humanos son capaces de construir taxonomías semánticas de manera eficiente. Sin embargo, la construcción manual de ontologías representa un trabajo intensivo, debido a que el conocimiento crece y es difícil generar, mantener y expandir con adecuada calidad ontologías de gran tamaño (Yeh, y otros, 2008). Por ello se hace necesario aplicar métodos que permitan disminuir el costo en tiempo y esfuerzo para su construcción y mantenimiento.

¹ Según la Real Academia Española un topónimo es un nombre propio de lugar.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

INTRODUCCIÓN

A partir de la situación problemática existente se plantea como **problema de investigación**: ¿Cómo disminuir el costo en tiempo y esfuerzo en el enriquecimiento de ontologías geográficas de gran tamaño?

El **objeto de estudio** de esta investigación lo constituye el enriquecimiento de ontologías, enmarcado en el **campo de acción** el enriquecimiento de ontologías geográficas.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar una herramienta de software que a partir del análisis de fuentes de información geográfica disponibles en Internet, permita disminuir el costo en tiempo y esfuerzo del enriquecimiento de ontologías geográficas de gran tamaño.

Desglosado en los siguientes **objetivos específicos**:

1. Realizar el marco teórico y el estado del arte correspondiente al enriquecimiento de ontologías geográficas, prestando especial interés en los métodos que utilizan fuentes de información geográfica disponibles en Internet.
2. Definir las fuentes de información geográfica disponibles en Internet a utilizar en la investigación.
3. Proponer un método semi-automático para enriquecer ontologías geográficas a partir de fuentes de información geográfica disponibles en Internet.
4. Implementar un plugin² para GENESIG que desarrolle el método propuesto.
5. Realizar pruebas de software a la herramienta desarrollada.

Se formularon las siguientes **preguntas científicas**:

1. ¿Cuáles son las bases que fundamentarán el estudio del marco teórico de la propuesta de solución según la bibliografía científica?
2. ¿Qué características definirá la propuesta de solución?
3. ¿Cómo realizar la extracción de propiedades, instancias y conceptos desde la fuente de información en internet hacia la ontología?
4. ¿Cómo integrar dos ontologías geográficas?

² Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

INTRODUCCIÓN

5. ¿Cuáles son las herramientas tecnológicas idóneas para darle solución al problema planteado y al mismo tiempo mantener la compatibilidad con GENESIG?

Para dar cumplimiento al problema planteado, se definieron las siguientes **tareas de la investigación**:

1. Indagación sobre los conceptos asociados al problema de la investigación.
2. Selección de las tecnologías y herramientas a utilizar en el proceso de desarrollo del sistema.
3. Realización del diseño de la solución propuesta.
4. Definición de los requisitos funcionales y no funcionales de la aplicación.
5. Confección del modelo de casos de uso de la aplicación.
6. Elaboración del modelo de diseño.
7. Implementación de las funcionalidades de la solución propuesta.
8. Diseño de los casos de prueba.
9. Realización de pruebas.

En correspondencia con el objetivo y las tareas propuestas se utilizaron los diferentes **métodos de la investigación**:

Métodos Teóricos

- **Analítico- Sintético:** Para descomponer el problema de la investigación en elementos por separados: Análisis e identificación de componentes ontológicos en la fuente de información e inserción de estos componentes en la ontología del dominio geográfico.
- **Histórico-Lógico:** En la primera etapa de la investigación se lleva a cabo el estudio crítico de investigaciones similares y un análisis de las mismas para un mejor entendimiento.

El presente trabajo está conformado, además de la introducción por 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

Capítulo 1: Fundamentación Teórica. En este capítulo se describe brevemente el problema, los principales conceptos y definiciones asociados con el dominio del problema. Se realiza un estudio relacionado con las ontologías geográficas y las fuentes de información disponibles en internet así como un análisis acerca de las soluciones existentes tanto a nivel nacional como internacional. También se argumenta la selección de herramientas y tecnologías a utilizar en el proceso de desarrollo de software, así como lenguaje de programación, metodologías a utilizar y entorno de desarrollo.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

INTRODUCCIÓN

Capítulo 2: Descripción de la propuesta de solución. Se realiza la propuesta de la solución. Se argumenta mediante el desarrollo del modelo de dominio. Se abordan también los requerimientos funcionales y no funcionales, la descripción de los casos de usos del sistema. Además se define la arquitectura de software a utilizar, así como los patrones y se da paso a la confección de los diagramas de clases del diseño.

Capítulo 3: Implementación y prueba. Se confeccionan los diagramas de componentes y despliegue. También se realizan las pruebas de software a la solución propuesta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se describen los conceptos asociados al dominio del problema, a través del campo de acción donde está enfocada la presente investigación, así como los aspectos y características generales relacionados con el tema de las ontologías y su aplicación en la web semántica. Se realiza una descripción del estado del arte, donde se exponen los aportes al conocimiento realizados por diferentes investigadores. Se identifican la metodología y herramientas utilizadas en el proceso de enriquecimiento de una ontología geográfica, los métodos empleados para lograr identificar atributos de la ontología, tecnologías, entre otros componentes utilizados en la elaboración de la solución.

1.1 Marco Teórico

Para el hombre, conservar el conocimiento es de gran importancia, por eso, desde los tiempos remotos ha intentado perpetuar ese conocimiento a través de la historia. Hoy en día es un problema representar ese conocimiento y tratar de adaptarlo de forma tal que los equipos de cómputo dispongan e interpreten la forma de pensar y el razonamiento de los humanos. En los últimos años existe un gran auge con respecto a la utilización de ontologías para la representación de conocimiento.

El termino ontología surgió de la filosofía. Los primeros pensantes alrededor del tema fueron los científicos Sócrates y Aristóteles, donde hacen distinguir la idea que “es la manera natural de las cosas y su forma de existencia”.

Por otra parte, (Gruber, 1993) precisa que “Una ontología no es más que una especificación explícita de una conceptualización”. Una conceptualización se refiere a un modelo abstracto de un fenómeno cualquiera en el mundo y los conceptos más relevantes del mismo. Por explícita se refiere a que los conceptos y las restricciones serán definidos de una forma entendible y precisa.

(Guarino, 1997) define a las ontologías como “Una teoría lógica justificando el significado deseado de un vocabulario formal, es decir, su propósito ontológico para una conceptualización particular del mundo”. En el contexto referente, una ontología puede solamente especificar una conceptualización de una manera débil.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

Teniendo en cuenta las anteriores definiciones, se arribará al concepto de ontología cuando al menos se esté en presencia de conceptos formales, propiedades, relaciones y reglas de restricción, existiendo entre los mismos, una interrelación.

(Guarino, 1998) demuestra que existen cuatro tipos de ontologías: ontología de alto nivel, ontología de dominio, ontología de tarea y ontología de aplicación. Las de alto nivel o nivel superior describen los conceptos generales como el espacio, tiempo, materia, objeto, evento, acción, los cuales son independientes de un problema o dominio en particular. Las de aplicación describen conceptos dependiendo de un dominio y de una tarea en particular, la cual es una especialización de ambas ontologías relacionadas (ontología de dominio y ontología de tarea). Y por último, las de dominio y de tareas, que describen, respectivamente, el vocabulario relacionado a un dominio genérico. En la presente investigación se hace uso de las ontologías de dominio, enmarcadas en el dominio geográfico.

En la Figura 1 se representan los cuatro tipos de ontologías según (Guarino, 1998).

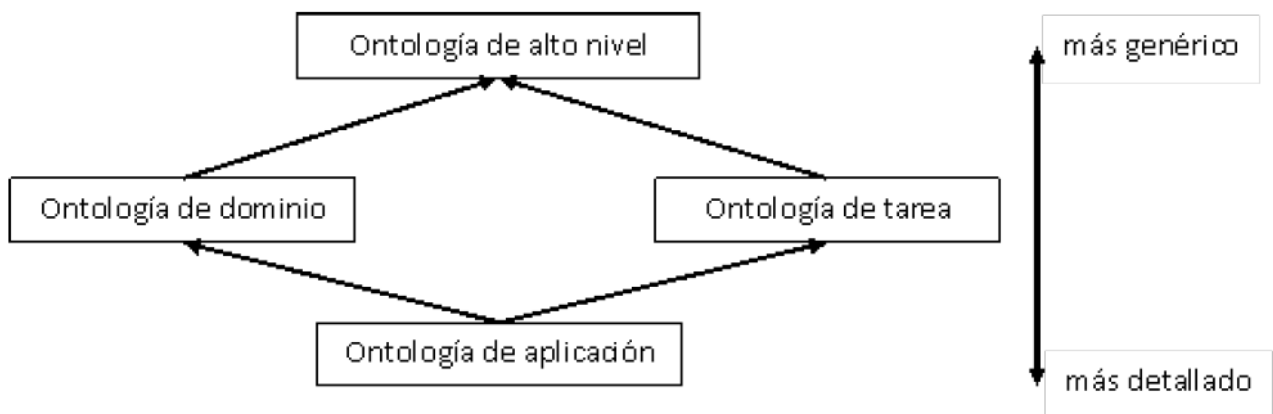


Figura 1: Tipos de ontologías.

Existen diversos autores que presentan definiciones sobre las ontologías, los argumentos expuestos por (Ehrig, 2007), definen que una ontología O es definida a través de la siguiente tupla:

$$O = (S, A, KB, Lex)$$

Consiste en:

- un núcleo de ontología S ,
- un sistema L-axioma A ,
- una base de conocimiento KB y

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- un lexicón Lex .

Donde:

Un núcleo de ontología es una estructura $S = (C, \leq_C, R, \leq_R)$ consistente en:

- dos conjuntos disjuntos C y R cuyos elementos son llamados identificadores de conceptos e identificadores de relaciones (o simplemente conceptos y relaciones).
- un orden parcial \leq_C en C , llamado jerarquía de conceptos o taxonomía.
- una función $\sigma: R \rightarrow C \times C$ llamada signatura, donde $\sigma(r) = \langle d(r), r(r) \rangle$ con r , dominio $d(r)$, y rango $r(r)$,
- un orden parcial \leq_R en R , llamado jerarquía de relaciones, donde $r_1 \leq_R r_2$ si y solo si $(r_1) \leq_C d(r_2)$ y $r(r_1) \leq_C r(r_2)$.

Un sistema L-axioma para un núcleo de ontología es un par $A = (A, \alpha)$ donde:

- A es un conjunto cuyos elementos son llamados identificadores de axiomas y
- $\alpha: A \rightarrow L$ es una asociación.

Los elementos de $A := \alpha(A)$ son llamados axiomas. S es considerada como parte del lenguaje L .

Una base de conocimiento es una estructura $K = (C, R, I, \iota_C, \iota_R)$ consistente en:

- dos conjuntos disjuntos C y R definidos anteriormente.
- un conjunto I cuyos elementos son llamados identificadores de instancias (o instancias).
- una función $\iota_C: C \rightarrow \mathfrak{P}(I)$ llamados instanciación de conceptos.
- una función $\iota_R: R \rightarrow \mathfrak{P}(I^2)$ con $\iota_R(r) \subseteq \iota_C(d(r)) \times \iota_C(r(r))$, para todo $r \in R$. La función ι_R es llamada instanciación de relaciones.

Un lexicón, para una ontología, es una estructura $L := (GC, GR, GI, RC, RR, RI)$ consistente en:

- tres conjuntos GC , GR , y GI cuyos elementos son llamados signos para conceptos, relaciones e instancias, respectivamente.
- una relación $RC \subseteq GC \times C$ llamado referencia léxica para conceptos, RR y RI análogamente.

Por otro lado (Euzenat, y otros, 2007) profundizan en la definición de ontología, detallándola como una tupla $O = \langle C, I, R, T, V, \leq, \perp, \in, = \rangle$, tal que:

- C es el conjunto de clases.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- I es el conjunto de individuos o instancias.
- R es el conjunto de relaciones.
- T es el conjunto de tipos de datos.
- V es el conjunto de valores (C, I, R, T, V siendo pares disjuntos).
- \leq es una relación en $(C \times C) \cup (R \times R) \cup (T \times T)$ llamado especialización.
- \perp es una relación en $(C \times C) \cup (R \times R) \cup (T \times T)$ llamada exclusión.
- \in es una relación sobre $(I \times C) \cup (V \times T)$ llamada instanciación.
- $=$ es una relación sobre $(I \times C) \times (I \cup V)$ llamada asignación.

Donde:

Las clases o conceptos son las entidades principales de una ontología. Son interpretadas como un conjunto de individuos en un conjunto. Son introducidas en OWL por la construcción *owl: Class*.

Los individuos, objetos o instancias son interpretados como un individuo particular de un dominio. Son introducidas en OWL por la construcción *owl: Thing*.

Las relaciones son la noción ideal de una relación independientemente al tipo que se aplique. Las relaciones son interpretadas como un subconjunto de producto del dominio. Son introducidas en OWL por *owl: ObjectProperty* u *owl: DatatypeProperty*.

Los tipos de datos son una parte particular del dominio que especifica valores. Opuestamente a los individuos, los valores no tienen identidades.

Los valores de datos son valores propiamente que un objeto puede tomar.

La especialización entre dos clases o dos propiedades es interpretada como la inclusión de las interpretaciones. La especialización es introducida por OWL por *rdfs: subclassOf* o *rdfs: subPropertyOf*.

La exclusión entre dos clases o dos propiedades es interpretada como la exclusión de sus interpretaciones, por ejemplo, cuando su intersección es vacía. La exclusión es introducida en OWL por *owl: disjointWith*.

La instanciación o tipado entre individuos y clases, instancias de propiedades y propiedades, valores y tipos de datos es interpretado como una membresía. La instanciación es interpretada en OWL con *owl: type*.

Sin embargo (Hess, 2008), plantea que una ontología puede ser definida como una 4-tupla

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

$O = \langle C, P, I, A \rangle$ donde:

- C es un conjunto de conceptos.
- P es un conjunto de propiedades.
- I es un conjunto de instancias.
- A es un conjunto de axiomas.

Un concepto C es cualquier fenómeno de interés a ser representado en la ontología. Una propiedad P es un componente que es asociado a un concepto C con el objetivo de caracterizarlo, pero es definido fuera del ámbito de un concepto. Puede ser una propiedad de tipo de dato, lo que significa que su valor es un tipo de dato, como string, integer, double, etc., o una propiedad de tipo objeto. La propiedad tipo de dato puede ser vista como un atributo de base de datos, mientras que la propiedad tipo de objeto puede ser vista como una relación de base de datos. Una instancia I es una ocurrencia de un concepto C , con un valor para cada propiedad P asociada al concepto y un único identificador. Un axioma A describe una relación jerárquica entre conceptos, o provee una asociación entre una propiedad y un concepto, o asocia una instancia con el concepto al cual pertenece, o define restricciones sobre las propiedades dentro del contexto de un concepto.

El concepto de ontología propuesto por Hess, lo expresa de la manera más breve y explícita posible. En él se abordan todos los atributos que debe poseer una ontología como mínimo para cumplir con su definición. Descarta las propiedades de relaciones, ya que en los axiomas encierra dicha cualidad.

(Ramos, y otros, 2007) mencionan que los lenguajes para codificar ontologías deben contemplar ciertos aspectos tales como: sintaxis bien definida, semántica específica, suficiente expresividad, fácilmente traducible entre los lenguajes ontológicos y permite eficiencia para realizar razonamientos. Entre los lenguajes para representar ontologías existen XML, RDF, OIL, DAML + OIL y OWL, de los cuales la *World Wide Web Consortium (W3C)* recomienda RDF y OWL.

El *Resource Description Framework (RDF)*, es una recomendación de la W3C para representar metadatos en la Web. Proporciona un medio para agregar semántica a un documento sin referirse a su estructura. RDF es una infraestructura para la codificación, intercambio y reutilización de metadatos estructurados.

RDF está pensado para situaciones en las que las necesidades de información en la Web necesiten ser procesadas por las aplicaciones. Proporciona un marco común para expresar esta información para que pueda ser intercambiada entre aplicaciones sin pérdida de significado. Dado que es un marco común,

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

los diseñadores de aplicaciones pueden aprovechar la disponibilidad de programas de análisis comunes RDF y herramientas de procesamiento (Schreiber, y otros, 2014).

Web Ontology Language (OWL), es un lenguaje de marcado semántico desarrollado por la W3C para publicar y compartir ontologías sobre la *World Wide Web*. Es una extensión del vocabulario de RDF y se deriva de *DAML+OIL*. OWL está diseñado para ser utilizado por aplicaciones que necesitan procesar el contenido de la información en lugar de solo presentarla a las personas (McGuinness, y otros, 2004). Existen diversas versiones de OWL, como *OWL Lite*, *OWL DL*, *OWL Full*, cada una de ellas incorpora diferentes funcionalidades. *OWL Full* es el más robusto, por lo que necesita mayor rendimiento computacional para inferir conocimiento (McGuinness, et al., 2004).

El lenguaje OWL es el más utilizado en el desarrollo de ontologías. Permite expresar una mayor claridad de las relaciones entre diferentes clases y especificación de propiedades, la versión *OWL Lite* posee rapidez y eficiencia, pero en su interior sus relaciones no son fuertes, por otra parte *OWL Full* contiene en su interior la capacidad de llevar a cabo relaciones entre los componentes de la ontología de manera sólida, pero la variable rendimiento mengua al ser estas sus características, por último se encuentra *OWL DL*, que obtiene propiedades de la rapidez y sencillez de *OWL Lite* y robustez de *OWL Full* (McGuinness, et al., 2004), por tanto como se desea ganar en expresividad, pero a la misma vez en rendimiento, se escogerá la versión *OWL DL*.

Entre los objetivos de las ontologías, (Guzmán, 2006) hace alusión que persiguen:

1. Compartir la comprensión común de la estructura de información entre personas o agentes de software.
2. Permitir la reutilización del conocimiento perteneciente a un dominio.
3. Permitir representar de manera explícita los supuestos de un dominio.
4. Separar el conocimiento de un dominio, del conocimiento que se pueda denominar operacional.
5. Permitir analizar el conocimiento de un campo.

Una geontología es una ontología que ofrece una descripción de entidades geográficas y difiere de otras ontologías por la presencia predominante de relaciones topológicas. Las geontologías pueden utilizarse para hacer explícita la semántica del contenido de la información geográfica de Servicios Web, con el fin de mejorar el descubrimiento y recuperación en la web (Lutz, M., & Klien, E, 2006).

Otro concepto lo aborda (Smith and Mark, 1998) quien plantea que “Una geontología es entonces diferente de otras ontologías debido a la topología y las relaciones espaciales, juegan un papel importante en el dominio geográfico. Además, los objetos geográficos pueden estar conectados o

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

contiguos, dispersos o separados, cerrados o abiertos, y por lo general son complejos y tienen partes constituyentes”.

En la definición de (Oliva-Santos, 2009), afirma que las geo-ontologías son estructuras en las que es posible la integración o asociación dato-conocimiento.

Las ontologías geográficas poseen las mismas características de las ontologías convencionales, la diferencia consiste en que estas poseen la peculiaridad de pertenecer al dominio geográfico y contienen relaciones topológicas entre objetos geoespaciales.

1.2 Fuentes de información

Existen diversas fuentes de información disponibles en internet de las cuales se pueden obtener resultados satisfactorios para el enriquecimiento de la ontología. Entre ellas, se analizarán las utilizadas con más frecuencia:

1.2.1 Centro de Excelencia para los Servicios de Información Geográfica

El Centro del Estudio Geológico americano de Excelencia para la Ciencia de Información Geoespacial (CEGIS, por sus siglas en inglés), se creó en 2006 y desde ese tiempo ha proporcionado la investigación principalmente en el apoyo de El Mapa Nacional.

- El Mapa Nacional ofrece una colección de pequeña escala de datos, originalmente desarrollada por la Edición 1997-2014 del Atlas Nacional de Los Estados Unidos, disponible para los usuarios.
- Se puede consultar, ver y descargar más de 200 000 mapas topográficos de Estudios Geológicos de Estados Unidos (USGS, por sus siglas en inglés).
- Para ver y transmitir datos que están disponibles en el dominio público, los fabricantes más experimentados del mapa y los usuarios de información geográficas profesionales, deben procesar el visor de mapa nacional.

Las presentaciones y publicaciones de los investigadores del CEGIS, documentan los logros de la investigación que incluyen los adelantos en el plan del mapa topográfico electrónico, la generalización, la integración de los datos, las proyecciones del mapa, el nivel del mar, la semántica geoespacial, la otología y la información geográfica (CEGIS, 2015).

CEGIS posee un plan de investigación 2013-2018, el cual incluye nuevas áreas de la investigación, ejemplo: modelos de datos, semántica del espacio geográfico, alto rendimiento computacional, información geográfica ofrecida, medios sociales de comunicación, integración de datos y representaciones de multiescala.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Los investigadores de CEGIS han desarrollado una ontología de alto nivel topográfico para el vector de las capas temáticas de El Mapa Nacional. La ontología mantiene las definiciones semánticas, recuperando y viendo vía web, las instancias particulares de las características topográficas básicas que están disponible en estas bases de datos. El trabajo continuado está desarrollando el plan de modelos de ontologías para definir y acceder a instancias de complejas características topográficas. Los modelos del plan de ontología pueden incorporar modelos extraídos de modelos del terreno digitales o relaciones integradas entre temas de los datos múltiples o resoluciones (CEGIS, 2015).

1.2.2 Wikipedia

La palabra Wikipedia es acuñada por su creador Larry Sanger en los inicios del año 2001. Es la contracción de wiki, una tecnología para desarrollar sitios web colaborativos, y tiene su origen de la palabra *wikiwiki*, que en el idioma hawaiano significa “rápido”.

Wikipedia es una enciclopedia libre, editada colaborativamente, cuya financiación está basada en donaciones. Posee más de 37 millones de artículos en 287 idiomas que han sido redactados en conjunto por personas voluntarias de todo el mundo, y cualquier persona con acceso al proyecto puede editarla. Se encuentra entre los 10 sitios web más populares del mundo y su éxito ha propiciado la aparición de proyectos hermanos como Wikcionario, Wikilibros, Wikiversidad, Wikisource, entre otros.

Ostenta características esenciales que definen su función en la web: (Wikipedia, 2015)

- Estructuración en el almacenamiento, la recopilación y la transmisión de la información.
- Posee contenido abierto, esto significa que permite explícitamente su copia.
- Es un wiki, por lo que puede ser editado por cualquiera.

Wikipedia es un proyecto dirigido a —literalmente— todo el mundo y editable por —literalmente— cualquier persona (Wikipedia, 2015).

1.2.3 GeoNames

GeoNames es un sitio web que utiliza topónimos, y donde cualquier usuario puede consultar servicios referentes a los datos geográficos tanto globales como administrativos, áreas, lugares poblados, datos sobre tiempo entre otros. Es gratuito y accesible desde internet (GeoNames, 2015).

Los topónimos se plantean cuando surge la necesidad de nombrar lugares nuevos o ausentes del repertorio toponímico tradicional. A ello se añaden los cambios de denominación impuestos por las nuevas realidades geopolíticas o reclamados por los gobiernos locales, que a menudo chocan con las denominaciones tradicionales. Muchos topónimos provienen, además, de lenguas que utilizan alfabetos

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

no latinos o que carecen de representación escrita, los cuales han de transcribirse y adaptarse siguiendo en lo posible los usos ortográficos propios, con el fin de que su grafía refleje adecuadamente su pronunciación.

(GeoNames, 2015) ofrece una amplia gama de servicios web sofisticados.

- La apreciación global de servicio web.
- La documentación de servicio web.
- Las bibliotecas clientes del servicio web.

El usuario puede acceder a la información libremente a través de una interfaz. Es posible encontrar topónimos a través de la búsqueda de un código postal o temperatura ambiente, además pueden consultarse enlaces en la Wikipedia del artículo en cuestión.

La Ontología de GeoNames hace posible agregar la información semántica geoespacial a la W3C. Presenta 8.3 millones de topónimos, GeoNames ahora tiene una dirección URL con un servicio web RDF correspondiente. Otros servicios describen la relación entre los topónimos.

1.2.4 Análisis de las fuentes de información

Presentadas las anteriores fuentes de información disponibles en internet y sus respectivas características y propiedades, se llega a la conclusión que el Centro de Excelencia para los Servicios de Información Geográfica posee información geoespacial y aspectos a tener en cuenta para la selección como la riqueza en la información de los objetos espaciales, pero no posee el nivel de popularidad ni la riqueza de datos como GeoNames o Wikipedia. Wikipedia por otro lado, se encuentra entre los sitios más populares y visitados diariamente, pero posee fuertes críticas ya que la información puede ser accesible y editada por cualquier persona, siendo esto una ventaja, pero también genera un problema, pues se puede introducir deliberadamente o no, información parcial. Por tanto existe el tema de la censura de artículos que no se ajustan a temas de interés por otras personas.

Por último, GeoNames reúne las principales características de ambos sitios, posee gran popularidad debido a la riqueza de información geográfica del sitio ya que contiene alrededor de 10 millones de nombres geográficos y sobre 9 millones de rasgos únicos de 2.8 millones de lugares poblados. Además, GeoNames supera los 150 millones de demandas de servicio por día. Por el anterior análisis, se arriba a la conclusión que la fuente de información que conserva más riqueza de información geoespacial y la más adecuada para tomar sus datos y procesarlos, es en este caso GeoNames.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.3 Estado del Arte

Las ontologías proporcionan una organización de conceptos en la representación de conocimientos que representa un marco de trabajo conceptual para el análisis o consulta de información de un dominio. Existe la necesidad de realizar tareas de razonamiento en ellas, por lo cual se deben integrar módulos o herramientas de razonamiento de acuerdo al desarrollo ontológico realizado. Con su utilización se logra almacenar la semántica de los datos contenidos dentro de los sistemas de información. La semántica se refiere al significado real de los datos, a la integración de nomenclatura y conceptualización de los mismos.

A través de los tiempos el hombre ha intentado optimizar su trabajo, lograr mejor eficacia y eficiencia en las herramientas que realiza y servicios que brinda. Por ello surge la necesidad de automatizar el proceso de enriquecimiento de ontologías, con el fin de lograr aumentar su calidad, reduciendo el costo en tiempo y esfuerzo.

Se necesitan las ontologías geográficas para lograr describir las propiedades semánticas de los objetos geográficos existentes en un SIG y extraer la información desde las fuentes de internet para así enriquecer a dicha ontología.

Existen varios autores interesados en el tema de generación de ontologías y el enriquecimiento automático de las mismas:

Los autores (Buscaldi, et al., 2006), en *Inferring Geographical Ontologies from Multiple Resources for Geographical Information Retrieval*, hacen alusión a que “*muchos de los documentos que se pueden encontrar en la World Wide Web incluyen algún tipo de información geográfica, a menudo de manera implícita. El uso de los recursos como diccionarios geográficos y ontologías geográficas pueden mejorar los resultados en recuperación de información geográfica. Desafortunadamente, la construcción de dichas ontologías es un proceso largo y laborioso; por lo tanto, la construcción de una ontología de una manera semi-automática explotando múltiples fuentes es una tarea interesante y útil.*”

Se construyó una ontología geográfica, realizando una integración de datos de diccionarios geográficos como *WordNet*, *Geonet Name Server (GNS)*, *Geographic Name Information System (GNIS)* y la enciclopedia libre Wikipedia. La ontología es implementada a través de la base de datos de Prolog, lo que facilita insertar nuevas relaciones o datos. Los diccionarios GNS y GNIS son de libre acceso y proporcionan gran riqueza de información geográfica sobre lugares del mundo. WordNet por otra parte, posee información geográfica bastante pequeña por la falta de información explícita en los conjuntos de

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

sinónimos agrupados o synsets. En Wikipedia solo pueden añadir únicamente los nombres que se incluyen en los títulos de los artículos, aunque la enciclopedia puede ser editada por cualquier persona.

Con dicho trabajo se cumplió en gran parte el objetivo de enriquecer de una forma semiautomática una ontología para la ampliación de los términos geográficos en un conjunto de documentos, utilizada como recurso para la GIR³. Pero la solución propuesta no resolvió los problemas de ambigüedad de los datos.

Otra investigación es la de (Ruiz, 2007), quien propone una metodología para proporcionar una serie de descripciones semánticas donde se encuentren reflejadas las relaciones y propiedades que describen y definen los objetos geográficos, extrayendo los elementos de una geontología. Al utilizar las ontologías, se refleja una conceptualización real a cada uno de los contextos los cuales describen de manera explícita la semántica de todos los elementos que componen el entorno. Torres define que una relación directa de una ontología con una conceptualización es a través de un lenguaje y dicho lenguaje utiliza un vocabulario para la interpretación. Lo que plantea con su técnica es que a partir de los objetos geográficos, se puede dar a conocer descripciones semánticas a través de las relaciones entre dichos objetos.

Otro hallazgo fue el trabajo de SPIRIT⁴ el cual es un proyecto de investigación que está dirigido al diseño e implementación de un motor de búsqueda para encontrar documentos y bases de datos en la web en relación con los lugares o regiones mencionados en una consulta (Jones, y otros, 2002). El proyecto presenta técnicas para extraer automáticamente cualquier contexto geográfico de documentos web, mapas digitales y registra seguidamente los recursos que utilizarán los motores de búsqueda web para la clasificación de recursos geográficos.

La relevancia de la investigación radica en la utilización de una ontología de dominio geográfico para el apoyo de la consulta generada por el usuario, la descripción de la información y la clasificación de la información recuperada. La ontología está basada en los conceptos de diccionarios y tesauros geográficos, lo cual contendrá tanto nombres administrativos formales como no formales, pero que son utilizados comúnmente. La extracción de metadatos se realiza a través de un análisis de datos textuales y se enriquece semánticamente mediante técnicas automáticas y/o semiautomáticas basadas en ontologías, para la generación de anotaciones geográficas.

³ Geographic Information Retrieval (Recuperación de información geográfica).

⁴ Spatially-Aware Information Retrieval on the Internet (Recuperación de Información Consciente-Espacial en Internet).

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Desde que SPIRIT fue enfocado a desarrollar aplicaciones globales, el factor de almacenamiento de la información es uno de los principales factores que afecta el desempeño del sistema. No han obtenido la solución de cómo derivar los datos geográficos de los varios recursos de los datos para poblar la ontología de SPIRIT. Asimismo no poseen algoritmos concretos para ser asignados a la implementación de las funciones de acceso, consulta y procesamiento de la información en la ontología.

En la investigación de (Martínez, 2008), se ha propuesto una vía de solución a la georreferenciación de ubicaciones a partir de su nombre, utilizando el servicio web de GeoNames. Desarrolla un Enterprise JavaBean (EJB, por sus siglas en inglés) en el IDE NetBeans, para resolver la latitud y longitud de cualquier ubicación en el mundo, descrita por el usuario que utilice la aplicación. Implementa un método el cual a partir de la cadena de texto entrada de la ubicación y el servicio de geo localización, obtiene la información deseada.

Realiza la tarea utilizando tres capas de implementación, implementación del EJB, la implementación del servicio web y la implementación del cliente de escritorio. En la lógica de negocio es donde se recibe la solicitud del usuario en conjunto con la cadena de texto de entrada. La capa del servicio web permite a los usuarios acceder al servicio de georreferenciación proveniente de la capa anterior. La última es la que permite al usuario consumir el servicio web creado en la capa de servicio web.

El trabajo desarrollado puede servir de base y guía para el estudio de la presente investigación, pero no se toma como un modelo a seguir ya que obliga al usuario a solo utilizar el método de búsqueda por su ubicación geográfica. También logra el objetivo de consumir el servicio de GeoNames, retornándolo a una interfaz de usuario en un panel de edición, pero no hace uso de la ontología de GeoNames ni realiza ningún proceso de enriquecimiento de ontología.

Por las investigaciones antes analizadas se llega a la conclusión que ninguna de estas formas de enriquecer ontologías son válidas para los propósitos de la presente investigación, por lo que se decide crear la herramienta que semiautomáticamente enriquezca una ontología geográfica a partir de GeoNames.

1.4 Herramientas para gestionar ontologías

A continuación se muestra un conjunto de herramientas utilizadas con el objetivo de gestionar ontologías, así como sus características principales. Estas herramientas no poseen las características necesarias para hacer uso en su totalidad en el desarrollo de la solución del presente trabajo:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.4.1 Chimaera

Esta herramienta permite crear y mantener ontologías en la web. Proporciona un ambiente distribuido para navegar, crear, editar, modificar y usar ontologías. Utiliza diferentes formatos para la carga de bases de conocimientos. Reorganiza taxonomías y resuelve conflictos de nombres y edición de términos. Facilita la combinación permitiendo al usuario subir ontologías a su espacio de trabajo (Barrera, y otros, 2012).

Actualmente la herramienta Chimaera no es explotada por los usuarios, ya que existen otros artefactos más completos y efectivos.

1.4.2 Ontolingua

Se puede describir como un proyecto que proporciona un ambiente colaborativo distribuido, para crear, editar y modificar ontologías, disponible desde un entorno web. El servidor Ontolingua contiene una librería de ontologías compartidas por los usuarios, que fomenta muy bien el principio de reutilización de ontologías (Navarro, 2012).

Ontolingua incluye una *Application Programming Interface* (API) para integrar ontologías del servidor con agentes preparados para Internet y su diseño modular, ofrece un conjunto de librerías que permiten a los usuarios ensamblar rápidamente una nueva ontología (Barrera, y otros, 2012).

El servidor soporta más de 150 usuarios activos, pero actualmente no existe una disponibilidad de la herramienta para descargarla, ni se encuentra en uso.

1.4.3 Protégé

La herramienta para la edición de ontologías más ampliamente difundida en la comunidad científica es Protégé, un ambiente para el desarrollo de sistemas basados en conocimiento que ha evolucionado hacia un conjunto de herramientas de propósito más general, de distribución libre y código abierto (Navarro, 2012).

Protégé en su núcleo, implementa un rico conjunto de estructuras de modelado de conocimiento y acciones que apoyan la creación, visualización y manipulación de ontologías en varios formatos de representación. Es una plataforma que puede ser extendida con gráficos, diagramas, componentes animados para acceder a aplicaciones embebidas en sistemas de bases de conocimientos (Barrera, y otros, 2012).

Protégé según (García, 2007), realiza las siguientes funciones:

- Generar distintos tipos de documentación sobre la ontología, tales como documentos en HTML o estadísticas sobre ontología.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- Es posible importar ontologías en diversos formatos a partir de bibliotecas de ontologías como DAML Y *Ontolingua Server*, entre otras.
- Es posible crear entornos de trabajo colaborativo, que facilitan la edición de la ontología por varios usuarios a la vez.

Al realizar un análisis de las herramientas antes expuestas, se arriba a la conclusión de que poseen peculiaridades similares a la solución que se desea desarrollar. Estas soluciones pueden ser tomadas como guía o escogidas para constituir un punto de referencia debido a las características que poseen. No se concibe la utilización de dichas soluciones, pues aunque brindan la posibilidad de crear, editar y utilizar ontologías, no logran enriquecer una ontología ni son adecuadas para el trabajo en paralelo con la plataforma GENESIG, dejando clara la importancia y necesidad de construir un nuevo plugin para la misma que permita enriquecer una ontología geográfica a partir de las fuentes de información disponibles en internet.

1.5 Metodología de desarrollo de software

En la etapa de desarrollo de software, es de gran importancia utilizar una metodología de desarrollo ya que viabiliza la eficacia y calidad del software. Una metodología de desarrollo de software no es más que un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de los sistemas. Los grupos de las metodologías se dividen en dos grupos: tradicionales y ágiles.

Las metodologías tradicionales centran su atención en llevar una completa documentación, planificación y procesos de todo el proyecto. Los resultados de los procesos no son siempre predecibles, existiendo altos costos al implementar un cambio. Las metodologías ágiles están orientadas a entornos variables. Exigen reducir los tiempos de desarrollo manteniendo una alta calidad. Su prioridad es satisfacer al cliente mediante tempranas y continuas entregas, dando más valor a la creación del producto de software funcional sin escribir exhaustivamente la documentación. Lo anterior tributa a una elevada simplificación sin renunciar al aseguramiento de la calidad del producto (Letelier, y otros, 2006). A continuación se refleja una comparación entre ambas metodologías.

1.5.1 Metodología tradicional

1. Más artefactos. El modelo es esencial.
2. Más roles, más específicos.
3. El cliente interactúa con el equipo de desarrollo mediante reuniones.
4. Aplicables a proyectos de cualquier tamaño. Suelen ser efectivas en proyectos grandes y con equipos posiblemente dispersos.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

5. Se promueve que la arquitectura se defina tempranamente en el proyecto.
6. Énfasis en la definición del proceso: roles, actividades y artefactos.
7. Se espera que no ocurran cambios de gran impacto durante el proyecto.

1.5.2 Metodología ágil

1. Pocos artefactos. El modelado es prescindible.
2. Pocos roles, más genéricos y flexibles.
3. El cliente es parte del equipo de desarrollo.
4. Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio.
5. La arquitectura se va definiendo y mejorando a lo largo del proyecto.
6. Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.
7. Se esperan cambios durante el proyecto.

Debido a que en la propuesta de solución son esperados ciertos cambios durante el desarrollo del proyecto, el cliente interactúa sistemáticamente con el equipo de desarrollo, la arquitectura se va definiendo a lo largo del proyecto, no existiendo un contrato tradicional y permitiendo reducir la generación de documentos y artefactos, se arriba a la conclusión de elegir un enfoque ágil de desarrollo de software.

1.5.3 Agile Unified Process (AUP)

Para el desarrollo del sistema se utilizó la metodología AUP. Su construcción está basada en la metodología tradicional *Rational Unified Process* (RUP, por sus siglas en inglés), pero por sus características y propiedades forma parte de la familia de las metodologías ágiles.

La metodología de desarrollo de software AUP, siguiendo las apreciaciones de (Cordero, 2011), posee principios que se basan en la alianza ágil:

Simplicidad: Se realiza un diseño sencillo concibiendo sólo la documentación indispensable.

Centrarse en actividades importantes: Enfoque en las actividades que poseen realmente importancia para el desarrollo del software. Todas las actividades no tienen el mismo valor en el desarrollo del proyecto.

Agilidad: La metodología AUP, aunque tiene como base una metodología tradicional, cumple con los principios del conjunto de metodologías ágiles.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Herramienta de la interdependencia: No tiene restricciones a la hora de utilizar herramientas. Aunque se recomienda utilizar las que mejor se adapten al trabajo, que a menudo son las herramientas simples o incluso de código abierto.

Edil personal: Las personas consultarán la documentación pertinente, tomando así lo que necesiten y consideren necesario en el momento indicado, buscando una orientación de alto nivel. La metodología AUP provee vínculos a muchos elementos de interés, pero no obliga a seguirlos.

Adaptar el producto: Un producto desarrollado con la metodología AUP es sencillo y fácil de manipular a través de cualquier herramienta que edite documentos en formato HTML. No se necesita una herramienta en especial para lograr interactuar con el mismo.

El ciclo de vida de la metodología AUP, está compuesta por las siguientes fases (Cordero, 2011):

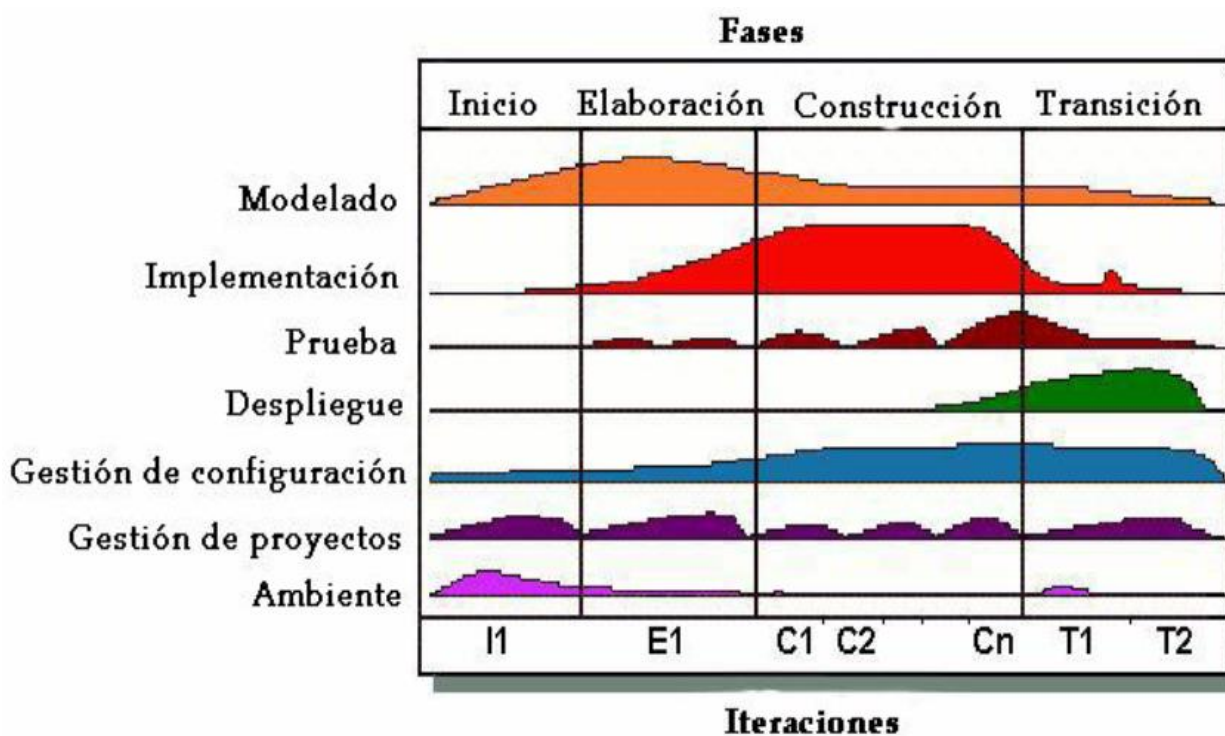


Figura 2: Disciplinas, fases e iteraciones de AUP.

(Cordero, 2011) Define las disciplinas de AUP como:

Modelado: Esta disciplina cumple como objetivo comprender el problema de dominio, el negocio que se aborda en el proyecto, así como llevar a cabo una propuesta de solución viable para resolver el problema de dominio.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Implementación: Esta disciplina cumple como objetivo transformar el modelo en un código ejecutable y realizar un nivel básico de pruebas.

Prueba: Esta disciplina cumple como objetivo realizar una evaluación objetiva para garantizar la calidad. Algunas de las actividades que realiza son: la búsqueda de defectos, la validación del sistema funcional como está establecido y verificación del cumplimiento de los requisitos.

Despliegue: Esta disciplina cumple como objetivo la prestación y ejecución del sistema y que el mismo esté a disposición de los usuarios finales.

Administración de la configuración: Esta disciplina cumple como objetivo gestionar el acceso a herramientas del proyecto. Incluye el seguimiento de las versiones, el control y la gestión de cambios

Administración o gerencia del proyecto: Esta disciplina cumple como objetivo dirigir las actividades que se llevan a cabo en el desarrollo del proyecto. Las actividades que incluye son: la gestión de riesgos, la dirección de personas, la coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurar que es entregado a tiempo y dentro del presupuesto requerido.

Entorno: Esta disciplina cumple como objetivo apoyar al resto de los esfuerzos por organizar que el proceso sea el adecuado, la orientación y herramientas estén disponibles para el equipo según sea necesario.

1.6 Herramientas y tecnologías

1.6.1 Visual Paradigm for UML 8.0

Visual Paradigm es una herramienta *Computer Aided Software Engineering* (CASE, por sus siglas en inglés) que emplea UML como lenguaje de modelado. Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, a través de la representación de todo tipo de diagramas. Proporciona la posibilidad de realizar diagramas de clases pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software (Pressman, 2008).

Entre las características de Visual Paradigm se encuentran:

- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Diseño centrado en Casos de Uso y enfocado al negocio, lo cual genera un software con mayor calidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- Genera código y realiza ingeniería inversa para diez lenguajes de programación, entre ellos Java, C++, PHP y XML Schema.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo, lo cual facilita considerablemente la comunicación.
- Disponibilidad en múltiples plataformas.
- Licencia: gratuita y comercial.
- Fácil de instalar y actualizar.

1.6.2 Lenguaje Unificado de Modelado 8.0

El Lenguaje Unificado de Modelado (UML) según (Larman, 1999), es el lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio. Entre las características y ventajas que posee este lenguaje se encuentran:

- Posee vocabularios y reglas que se centran en la representación conceptual y física de un sistema, para crear y leer modelos bien formados y además constituyen los planos de un sistema de software.
- Se basa en una notación gráfica y concisa y fácil de aprender y utilizar.
- Es un lenguaje de modelado visual de propósito general orientado a objetos.
- Agrupa notaciones y conceptos provenientes de distintos tipos de métodos orientados a objetos.

UML es utilizado en la presente investigación con el fin de visualizar, construir y documentar un sistema de software. Por tanto brinda un estándar para describir un plano del sistema y aspectos concretos del mismo.

1.6.3 Lenguaje de programación del lado del servidor

El lenguaje de programación del lado del servidor es aquel que es reconocido, interpretado y ejecutado por el servidor, el cual el cliente puede acceder y comprender la información contenida. Entre el conjunto de acciones de mayor importancia realizadas por un servidor, se encuentran el tratamiento de la información, el acceso a bases de datos entre otras.

Java 8.1

Se utiliza el lenguaje de programación Java. Es un lenguaje orientado a objetos y cumple, según (Marcos, 2012) con las siguientes ventajas:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Multiplataforma ya que el mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java. Esta es una de las principales características que favorece el crecimiento y difusión del lenguaje.

Interpretado y compilado a la vez: genera ficheros de clases compiladas, pero estas clases compiladas, son en realidad interpretadas por la máquina virtual de java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.

Seguro: La máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets, establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

Robusto: Exhaustivo de la fiabilidad. Java pone mucho énfasis en el control temprano de posibles errores, como compiladores de Java son capaces de detectar muchos problemas que en primer lugar aparecen durante el tiempo de ejecución en otros idiomas.

Multiproceso: Hoy día ya se ven terriblemente limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

El lenguaje Java, es entre todos el más utilizado en el trabajo con las ontologías ya que realiza sus actividades en paralelo con la librería Jena. Se decidió utilizarlo además, debido a que permite desarrollar aplicaciones bajo el esquema cliente-servidor y ejecutar tareas de forma simultánea entre ordenadores.

PHP 5.0

Para el desarrollo de la aplicación se utilizó además el lenguaje *HyperText Preprocessor* (PHP, por sus siglas en inglés), el cual es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML (Murugarren, 2011).

Dentro de las principales características definidas por (Achour, y otros, 2006) se encuentran:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- El código PHP, el cual generalmente define la lógica de funcionamiento interno de un sistema, es invisible al navegador web pues el intérprete PHP es encargado de ejecutarlo y enviar solo el resultado HTML al cliente, esto hace que la programación sea segura y confiable.
- Posee extensa y detallada documentación la cual puede ser consultada desde su página web.
- Implementa el paradigma orientado a objetos mediante la utilización de las clases las cuales poseen propiedades y métodos y pueden ser instanciadas.
- Posee capacidad de conexión con varios Gestores de Bases de Datos como son MYSQL y PostgreSQL.

1.6.4 Lenguaje de programación del lado del cliente

Por otro lado la programación del lado del cliente es la encargada de realizar una validación de los datos antes que el servidor los procese además de aportar dinamismo a la aplicación a la hora de mostrarla en un navegador web. En esta investigación se utiliza la librería ExtJS, basada en lenguaje JavaScript.

ExtJS 3.0

EXT JS es un conjunto de librerías *JavaScript* utilizadas para el desarrollo de aplicaciones web donde existe interacción con el cliente. Utiliza tecnologías como AJAX, DHTML y DOM. Es utilizada fundamentalmente para crear aplicaciones que poseen una gran comunicación con el usuario, el servidor procesa la información en segundo plano, realizando el proceso de búsqueda de resultados transparente al cliente. Por tanto, solicita, procesa y envía la información recibida en tiempo real, lo que permite a los programadores enfocar su trabajo en el desarrollo de las funcionalidades de la aplicación.

1.6.5 IDE NetBeans 8.0

Para el desarrollo de la aplicación se utilizó el *Integrated Development Environment* (IDE) NetBeans en su versión 8.0. Es una herramienta realizada principalmente para el trabajo con el lenguaje Java. Es un producto libre y gratuito sin restricciones de uso. Su misión consiste en evitar tareas repetitivas, facilitar la escritura correcta de código, disminuir el tiempo de depuración e incrementar la productividad del desarrollador. Cuenta con un depurador, perfilador de integración, herramientas para refactorizaciones, completamiento de código y control de versiones de archivos.

Este IDE está disponible para sistemas operáticos como Windows, Linux, Mac y Solaris. Posee código abierto y una plataforma de desarrollo que permite a los programadores crear en un corto período de tiempo aplicaciones de escritorio, web y móviles, utilizando la plataforma Java, PHP, AJAX, C y C++, JavaScript y Groovy y Grails.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

La plataforma NetBeans es una herramienta realizada exclusivamente para desarrolladores, con el fin de ayudar a escribir, compilar, depurar y ejecutar programas. Permite así, el desarrollo de aplicaciones utilizando en conjunto componentes de software.

Entre las principales características que posee el IDE se encuentran la administración de configuraciones de usuario, del almacenamiento, de las interfaces de usuario y de ventanas.

1.6.6 WSDL

Un *Web Services Description Language* (WSDL) es el estándar definido por el W3C para describir un servicio Web y crear el contrato entre el cliente y el proveedor (Balani, y otros, 2009).

La forma más común de exponer los servicios se basa en el estándar WSDL, documento XML que funciona como contrato de servicio. El uso de un repositorio permite disponer de los servicios organizados y fácilmente accesibles para ser reutilizados (Díaz, 2012).

1.6.7 Servidor Web Glassfish 4.0

Para la publicación de la WSDL de cada servicio web, se debe utilizar servidor web que se encargue de realizar esta tarea. El *NetBeans* en su instalación inicial, permite la elección de instalar el *Apache TomCat* y/o el *Oracle Glassfish Server*, servidores por defecto que vienen con la instalación.

Glassfish proporciona un servidor modular de peso ligero para el desarrollo de aplicaciones de la Plataforma *Java Enterprise Edition* (Java EE) y *Java Web Services* (JWS). Ofrece un rendimiento empresarial, escalabilidad y fiabilidad. *Glassfish* posee diecisiete puertos disponibles, solo que en sus inicios configura el puerto 8080 (conexión HTTP), para su uso (Oracle, 2015).

1.6.8 Apache Jena 2.12.0

Apache Jena es un *framework* java de código abierto basadas en ontologías, para construir aplicaciones.

En el año 2000 tuvo su desarrollo, en los laboratorios de *Hewlett Packard* (HP) y en 2009 HP cedió el proyecto a la fundación Apache, que decidió adoptarlo en noviembre de 2010.

Su Arquitectura incluye:

- API para trabajar (leer, procesar, escribir) ontologías RDF y OWL.
- Motor de inferencia para razonar sobre ontologías RDF y OWL.
- Estrategias de almacenamiento flexible para almacenar tripletas RDF en memoria o fichero.
- Motor de queries compatible con especificación SPARQL.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.6.9 GeoNames-Source 1.1.12

La librería facilita el acceso a los servicios web de GeoNames a través del lenguaje java. Posee la contrariedad que debe utilizar otra librería llamada jdom v1.0 la cual provee una solución completa para acceder, manipular y devolver datos XML para código java.

1.7 GENESIG

GENESIG es una plataforma tecnológica, desarrollada sobre software libre y encaminado a la creación de Sistemas de Información Geográfica en la web. Posee funcionalidades capaces de brindar georreferenciación y localización de recursos, además de ubicar objetos en el mapa que resulten de algún interés. El objetivo principal de la plataforma es agilizar el desarrollo de los SIG en la web, a partir de la reutilización de módulos y funcionalidades.

A continuación, los autores (Martínez, y otros, 2012) hacen alusión a algunas de las más complejas funcionalidades de GENESIG.

- Análisis de rutas con puntos predefinidos.
- Análisis de perfiles de alturas en el terreno.
- Búsqueda por direcciones postales.
- Tematiza información en el mapa por diversos criterios: corocromático y coropleta, gráficas dinámicas y símbolo proporcional.
- Generador de consultas dinámicas que permite editar y construir nuevas consultas en tiempo de ejecución.
- Consume capas de información geográfica en diferentes estándares: *PotsGIS* y *WMS2*.
- Exportar al "formato pdf" la sección el mapa que se está visualizando.

GENESIG es un sistema de software con una arquitectura basada en componentes, por tanto se puede afirmar que la plataforma soberana contiene características como la reusabilidad, la modularidad y la compatibilidad. En la arquitectura de componentes también predomina la robustez, debido a que los componentes deben operar en entornos heterogéneos. Su premisa es que los componentes cumplan con la alta cohesión y bajo acoplamiento.

1.7.1 Herramientas y tecnologías de la plataforma GENESIG

La plataforma soberana de desarrollo web GENESIG, esta fomentada sobre software libre, lo cual añade un valor importante al enfocarse en la creación de los componentes utilizando código abierto y gratuito.

Las tecnologías utilizadas fueron:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- El *Framework* de desarrollo de SIG CartoWeb v3.5.0.
- Servidor de mapas *MapServer*, utilizando la versión 5.4.1.
- El lenguaje de desarrollo dinámico del cliente *ExtJS v3.0* con *JavaScript*.
- Lenguaje *PHP v5.0* para su utilización en el lado del servidor, aprovechando la característica que es un lenguaje orientado a objetos.
- La extensión de *PostgreSQL*, *PgRouting v1.0.2* para el análisis de rutas.
- *PostgreSQL v8.3* y *PotsGIS v1.3* lenguaje de consultas para datos utilizado especialmente en las consultas espaciales y las vistas.
- Por último el servidor web Apache, en su versión 2.2.

1.7.2 Servidor Web Apache 2.2

El servidor web utilizado por la plataforma por su gran robustez, seguridad y configuración es el Apache. En la actualidad es el servidor con mejores y mayores resultados en el mundo web. Posee código abierto, es multiplataforma, fácil de utilizar y extensible, por que alcanza mayor popularidad en el mercado.

1.7.3 Arquitectura de GENESIG

La arquitectura de GENESIG es una arquitectura distribuida, basada en componentes y orientada a servicios, lo cual da lugar a brindar altos niveles de escalabilidad facilitando su desarrollo y mantenimiento.

A la plataforma se le integrará el módulo que se está desarrollando en la presente investigación, por lo tanto se tiene en cuenta las herramientas utilizadas por la plataforma para lograr una efectiva compatibilidad con la misma.

Conclusiones del capítulo

Al concluir este capítulo, se lograron analizar explicaciones de los términos referentes a la investigación. Son presentadas algunas de las principales soluciones existentes a nivel nacional e internacional, identificándose así un grupo de herramientas óptimas para resolver el problema científico y mantener la compatibilidad con GENESIG. Se realiza un estudio de las metodologías, herramientas y tecnología a utilizar, permitiendo escoger de manera correcta una vía de solución a la necesidad de desarrollar un sistema que corrija los problemas existentes.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se describirá el proceso llevado a cabo durante la fase de análisis, diseño e implementación de solución siguiendo un enfoque teórico, donde se analiza la solución. Se realiza la descripción de la propuesta de solución. Se describe el diagrama de casos de uso, del negocio, el modelo de dominio según las características presentadas por la metodología AUP, así como los requisitos funcionales y no funcionales que debe cumplir el software.

2.1 Propuesta de solución

La propuesta de solución se divide en cinco partes fundamentales, que serán explicados detalladamente a continuación:

1. Consumo de servicios web de GeoNames.

Primero se realiza la conexión al sitio de GeoNames, para que acepte la salida de los métodos implementados y pueda devolver respuestas a la consulta realizada por el usuario. Se implementa un método a través del cual se obtiene la ontología de datos geográficos que contiene el sitio. La ontología actual de GeoNames (*ontology_v3.1.rdf*), no contiene ninguna instancia del concepto lugar.

2. Insertar los componentes ontológicos en la ontología de GeoNames.

Al leer la ontología de GeoNames, apoyando el trabajo en la librería apache jena, se crea un concepto nombrado places (lugares) y se añaden como instancia, los topónimos resultantes de la consulta realizada por el usuario. Los topónimos se obtienen a través del consumo de servicios web que posee la librería GeoNames source, brindada por el propio sitio. En la ontología, se insertarán conceptos, instancias, propiedades de datos y de objetos con respecto a cada topónimo resultante. Para esto, se implementa un método el cual tiene como función escribir en el modelo ontológico los resultados obtenidos cada vez que se realice la llamada a un servicio web.

3. Integración de la ontología del dominio geográfico con la ontología de GeoNames por medios automáticos.

Se utiliza una ontología básica del dominio geográfico creada anteriormente, ya que sirve de base para ontologías de menor nivel. El objetivo principal que se persigue es que a partir de la interacción entre el mapa y la geo-ontología, se obtendrán mejores resultados en la generación de consultas visuales por parte del usuario final (Martínez, y otros, 2015). La integración de ontologías es una técnica muy utilizada

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

ya que permite agregar nuevos componentes ontológicos. Consiste en la unificación de dos ontologías del dominio geográfico para obtener una con mayor riqueza semántica en los datos. En la presente investigación denota como integración de dos ontologías a la fusión de las mismas, con la excepción que se realiza un análisis sintáctico de los elementos de la ontología. Por tanto se implementa un método que cumple con lo antes descrito. Así pues, la automatización de la integración de las ontologías, implicaría una reducción en cuanto al costo en tiempo y esfuerzo en la construcción de ontologías de gran tamaño.

4. Refinamiento de la ontología resultante por parte del usuario final.

En este método interviene el usuario en la selección de los componentes ontológicos que desee agregar, modificar y eliminar. Por otro lado, la intervención del usuario en su construcción da lugar a la semiautomatización de la tarea, permitiendo al usuario tomar decisiones sobre la respuesta final.

5. Almacenamiento de la ontología resultante.

Por último, se almacenará la ontología resultante para su posterior uso.

Para lograr obtener una respuesta en la aplicación de la plataforma de desarrollo, la cual utiliza en su código fuente lenguaje *JavaScript* y *PHP*, existen diversas variantes de soluciones, la utilizada en la presente investigación es la creación de la *wSDL* de cada servicio de *GeoNames*, para así lograr solicitar los datos de cada servicio web y devolver la respuesta en formato *JSON*.

2.2 Modelo de dominio

El Modelo de Dominio permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del problema. Contiene las clases conceptuales del mundo real. Puede utilizarse para obtener y expresar el entendimiento de un proceso en análisis como antesala del diseño de un sistema ya sea de software o de otro tipo. El Modelo de Dominio se presenta en forma de diagrama de clases donde se exponen los principales conceptos y roles del sistema en cuestión (Rumbaugh, 2000).

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

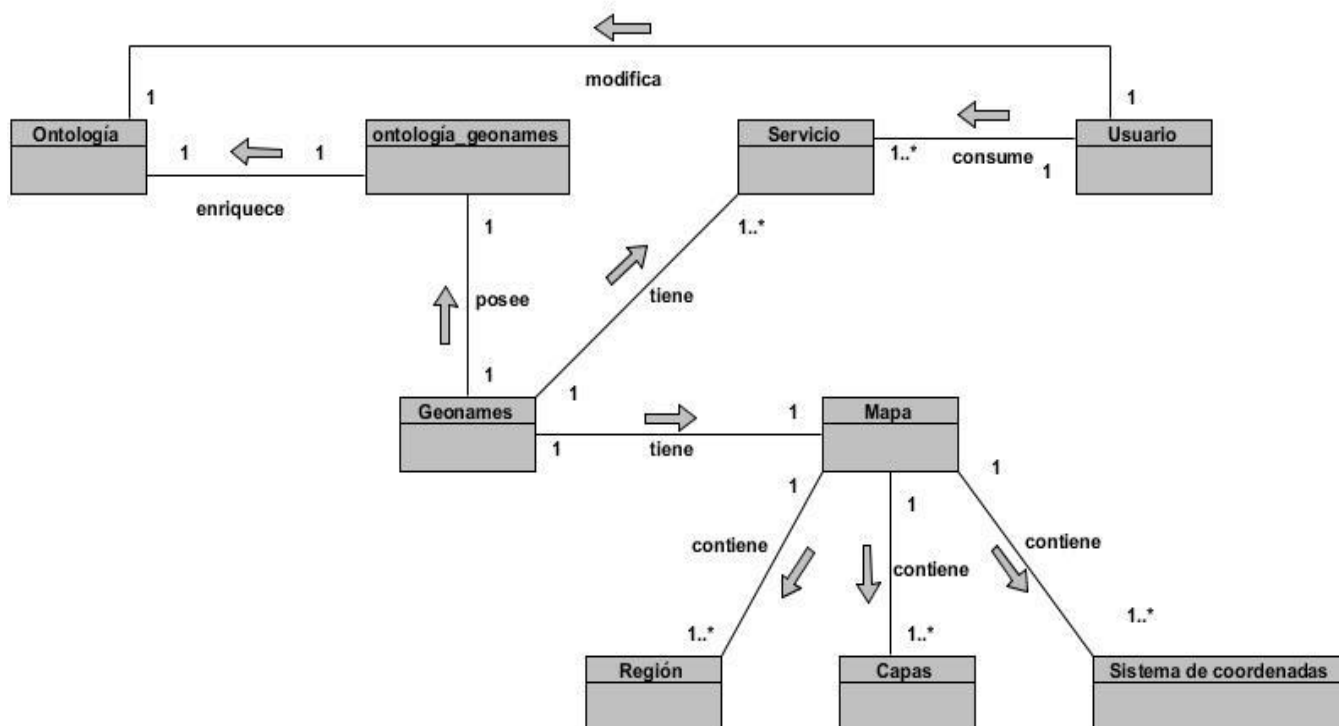


Figura 3: Modelo de dominio.

Descripción del Modelo de dominio

Un usuario consume uno o varios servicios web del sitio *GeoNames*. *GeoNames* y utiliza un Mapa que a su vez contiene Regiones, Capas y un Sistema de coordenadas. Además posee una ontología (Ontología_GeoNames) que es la encargada de almacenar los topónimos extraídos y enriquecer la ontología final (Ontología).

2.3 Especificación de requisitos

El sistema debe cumplir con una serie de requisitos funcionales y no funcionales (condiciones) para satisfacer las funcionalidades que requiere el sistema.

2.3.1. Requisitos funcionales

Los requisitos funcionales establecen el comportamiento del sistema y muestran a su vez cómo los casos de uso serán llevados a la práctica. Son además condiciones o capacidades que el sistema debe cumplir (Rumbauhg, 2000). A continuación se abordan los requerimientos definidos para el desarrollo del sistema que se visualiza en la plataforma GENESIG los cuales pueden ser modificados por el usuario:

RF1: Enriquecer ontología: Se añadirán componentes ontológicos a la ontología.

RF2: Gestionar conceptos.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

RF2.1: Insertar un concepto: El sistema debe permitirle al usuario insertar un nuevo concepto.

RF2.2: Modificar un concepto: El sistema debe permitirle al usuario modificar un concepto.

RF2.3: Eliminar un concepto: El sistema debe permitirle al usuario eliminar un concepto existente.

RF3: Gestionar propiedades.

RF3.1: Insertar una propiedad: El sistema debe permitirle al usuario insertar una nueva propiedad.

RF3.2: Modificar una propiedad. El sistema debe permitirle al usuario modificar una propiedad existente.

RF3.3: Eliminar una propiedad: El sistema debe permitirle al usuario eliminar una propiedad existente.

RF4: Gestionar instancias.

RF4.1: Insertar una instancia: El sistema debe permitirle al usuario insertar una nueva instancia.

RF4.2: Modificar una instancia: El sistema debe permitirle al usuario modificar una instancia existente.

RF4.3: Eliminar una instancia: El sistema debe permitirle al usuario eliminar una instancia existente.

RF5: Guardar ontología: Se almacena la ontología para su posterior uso.

2.3.2. Requisitos no funcionales

Los requisitos no funcionales son las restricciones de los servicios o funciones ofrecidas por el sistema, restringen el espacio de posibles soluciones (Sommerville, 2005), son las características que el sistema debe tener para el correcto desarrollo y despliegue del componente. A continuación se definen los requisitos no funcionales:

RNF1: Usabilidad.

- Proporciona un diseño con orden entendible.
- La información se mostrará estructuradamente y seguirá una forma lógica.

RNF2: Eficiencia.

- El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información, mayor será el tiempo de procesamiento. Al igual pasa con el tiempo de

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

respuesta, la velocidad de procesamiento de los datos, la actualización y recuperación dependerán de la cantidad de información que el sistema deba procesar.

RNF3: Software.

- Servidor: Sistema operativo *GNU/Linux Ubuntu Server 12.04.*, Servidor Web *Apache 2.0*, con módulo PHP 5 configurado con la extensión *pgsql* incluida, *PostgreSQL 8.4*, como Sistema Gestor de Base de Datos, *PostGIS* como soporte de datos espaciales, *MapServer 6.0* con extensión *PHP MapScript*.
- Cliente: Sistema Operativo *Ubuntu 12.04 LTS*, navegador *Mozilla Firefox/ Google Chrome*.

RNF4: Hardware.

- Servidor: Memoria RAM 2Gb, Disco duro 40Gb como mínimo, Tarjeta de red *Ethernet 10/100Mbps*.
- Cliente: Memoria RAM 256Gb como mínimo, Disco duro 40Gb, Tarjeta de red *Ethernet 10/100Mbps*.

2.4 Descripción de la solución propuesta

2.4.1 Diagrama de Caso de Uso

En la siguiente figura se puede apreciar el actor externo o usuario que es el encargado de iniciar la actividad del sistema, con los casos de uso presentes en el desarrollo del software.

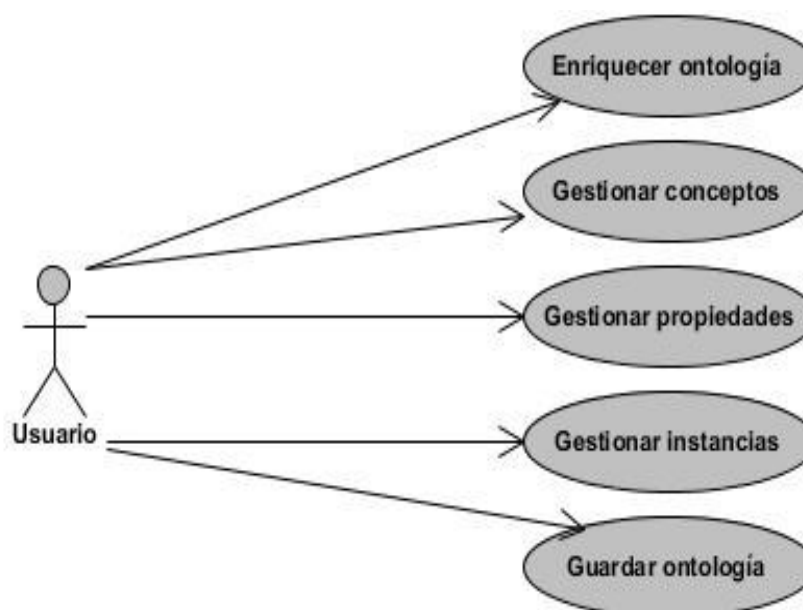


Figura 4: Diagrama de Casos de uso.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

Descripción de los actores del sistema

Un actor es un agente externo que interactúa con el sistema con vista a obtener un resultado esperado. Los actores que se especifican a continuación son con los que cuenta el sistema:

Tabla 1: Actores del Sistema.

Actor	Descripción
Usuario	Es aquel que tendrá acceso a las funcionalidades del sistema.

2.4.2 Priorización de casos de uso

Existen diversas clasificaciones para los casos de uso que se desarrollarán en la construcción de un software:

Críticos: Más importantes para los usuarios porque cubren las principales tareas o funciones que el sistema ha de realizar. Definen la arquitectura básica.

Secundarios: Sirven de apoyo a los casos de uso críticos, involucran funciones secundarias y tienen un impacto más modesto sobre la arquitectura, pero deben implementarse pronto porque responden a requerimientos de interés para los usuarios.

Auxiliares: No son claves para la arquitectura y completan casos de uso crítico o secundario.

Opcionales: Responden a funcionalidades que pueden o no estar en la aplicación, pero que no son imprescindibles en las primeras versiones.

En la presente investigación se encuentra presente en la clasificación de los casos de uso críticos, enriquecer ontología, en la clasificación de secundarios los casos de uso gestionar conceptos, gestionar propiedades, gestionar instancias y guardar ontología.

2.4.3 Especificación de los casos de uso

A continuación se muestra en la Tabla 2 la especificación de casos de uso enriquecer ontología, donde se describe con exactitud las acciones a realizar por el usuario y por el sistema al iniciar el caso de uso. El resto de las especificaciones de casos de uso se encuentran en el Anexo 1.

Tabla 2: Descripción del Caso de uso Enriquecer ontología.

Objetivo	El objetivo del CU enriquecer ontología es insertar componentes ontológicos.
Actores	Usuario.

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

Resumen	El CU inicia cuando el usuario selecciona un continente, el sistema valida los datos insertados en los casos que corresponda. El CU termina cuando se crea la ontología.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir conexión con internet.	
Poscondiciones	Creación de la ontología con éxito.	
Flujo de eventos		
Flujo básico Enriquecer ontología		
	Actor	Sistema
1.	Selecciona la opción de escoger continente.	
2.		Muestra una lista con los continentes y permite escoger uno de ellos.
3.	Selecciona el continente que desea.	
4.		El sistema permite escoger una de las siguientes opciones: -Actualizar tabla. - Escoger "País" del continente seleccionado. Ver Sección 1: Escoger país. - Filtrar. Ver Sección 2: Filtrar. - Unir ontologías. Ver Sección 3: Unir ontologías.
5.	Elige la opción "Actualizar tabla".	
6.		Muestra la tabla con los datos del continente correspondiente.
7.	Selecciona el botón Crear ontología.	
8.		Muestra en una ventana, permitiendo sobrescribir el archivo o no.

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

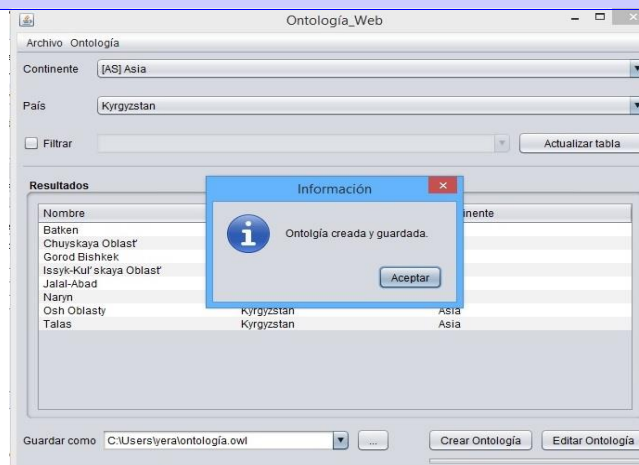
9.	Selecciona la opción “Si”.	
10.		Inserta los componentes ontológicos y muestra un mensaje de notificación “Ontología creada y guardada”.

Flujos alternos

9a Evento Selecciona la opción “No”.

	Actor	Sistema
10a		Cancela la operación.

Prototipo de interfaz



Sección 1: “Escoger país”

Flujo básico Escoger país.

	Actor	Sistema
4.	Selecciona el país deseado.	
5.		Muestra los datos del país seleccionado en la tabla.
6.	Oprime el botón Crear ontología.	
7.		Muestra en una ventana, permitiendo sobrescribir el archivo o no.
8.	Selecciona la opción “Si”.	

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

9.		Inserta los componentes ontológicos y muestra un mensaje de notificación "Ontología creada y guardada".
Flujos alternos		
6a Evento Selecciona la opción "No".		
	Actor	Sistema
7a		Cancela la operación.
Sección 2: "Filtrar"		
Flujo básico Filtrar.		
	Actor	Sistema
4.	Selecciona la opción "Filtrar".	
5.		Habilita el campo de texto permitiendo al usuario introducir el criterio de búsqueda.
6.	Introduce el criterio de búsqueda.	
7.	Elige la opción "Actualizar tabla".	
8.		Muestra la tabla con los datos de la consulta correspondiente.
9.	Oprime el botón Crear ontología.	
10.		Muestra en una ventana, permitiendo sobrescribir el archivo o no.
11.	Selecciona la opción "Si".	
12.		Inserta los componentes ontológicos y muestra un mensaje de notificación "Ontología creada y guardada".
Flujos alternos		
9a Evento Selecciona la opción "No".		
	Actor	Sistema
10a		Cancela la operación.

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

Sección 3: “Unir ontologías”		
Flujo básico Unir ontologías		
	Actor	Sistema
8.	Selecciona la opción “Unir ontologías”.	
9.		Muestra la ventana “Abrir”, permitiendo escoger el archivo a unir.
10.	Selecciona el archivo deseado.	
11.	Selecciona la opción “Abrir”.	
12.		Muestra otra ventana para almacenar el archivo.
13.	Selecciona la opción “Guardar”.	
14.		Muestra en una ventana, permitiendo sobrescribir el archivo o no.
15.	Selecciona la opción “Si”.	
15.		Une las ontologías seleccionadas.
Flujos alternos		
13a Evento Selecciona la opción “Guardar”.		
	Actor	Sistema
14a		Une las ontologías seleccionadas. Termina el CU.
Flujos alternos		
15a Evento Selecciona la opción “No”.		
	Actor	Sistema
16a		Cancela la operación.
Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.
Requisitos no funcionales	No procede.	

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

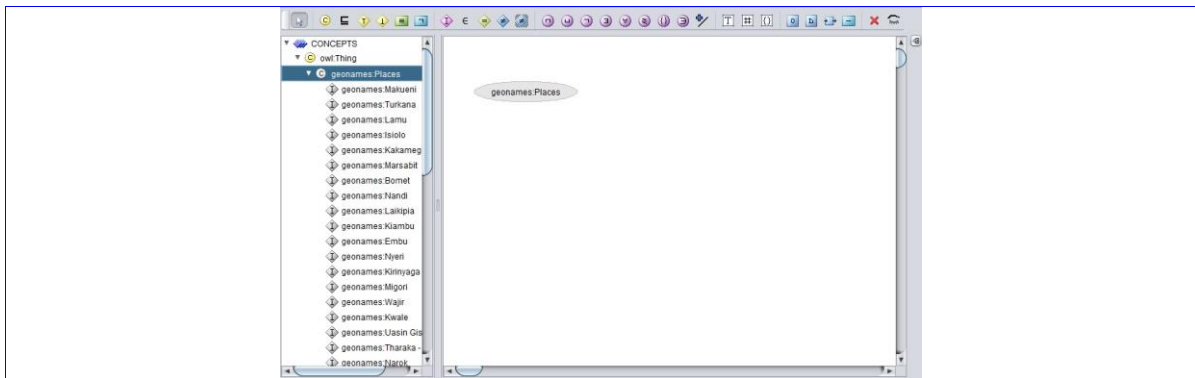
Asuntos pendientes	No procede.
---------------------------	-------------

Tabla 3: Descripción del Caso de uso Gestionar conceptos.

Objetivo	El objetivo de la gestión de conceptos es añadir, modificar y eliminar conceptos de la ontología.	
Actores	Usuario.	
Resumen	El CU inicia cuando el usuario selecciona un concepto de la ontología. El CU termina cuando se inserta, modifica o elimina un concepto.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Abrir el editor de ontologías.	
Poscondiciones	Se inserta, modifica o elimina un concepto de la ontología exitosamente.	
Flujo de eventos		
Flujo básico Gestionar conceptos		
	Actor	Sistema
1.	Selecciona la opción de abrir el editor de ontologías.	
2.		Muestra una interfaz que permite realizar las siguientes opciones: - Insertar un concepto. - Modificar un concepto. Ver Sección 1: "Modificar concepto". - Eliminar un concepto. Ver Sección 2: "Eliminar concepto".
3.	Selecciona el concepto a insertar.	
4.	Arrastra el concepto al panel principal.	
5.		Muestra los conceptos en el panel principal. Termina el CU.
Prototipo de interfaz		

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.



Sección 1: “Modificar concepto”

Flujo básico Modificar

	Actor	Sistema
2.	Inserta un nuevo concepto.	
3.	Selecciona el concepto.	
4.	Elige la opción modificar.	
5.		Permite la modificación del nombre del concepto.
6.	Escribe el nombre deseado.	
7.		Actualiza el nombre y el listado de conceptos. Termina el CU.

Sección 2: “Eliminar concepto”

Flujo básico Eliminar concepto

	Actor	Sistema
2.	Selecciona un concepto del panel principal.	
3.		Marca el concepto seleccionado.
4.	Elige la opción eliminar.	
5.		Elimina el concepto. Termina el CU.

Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

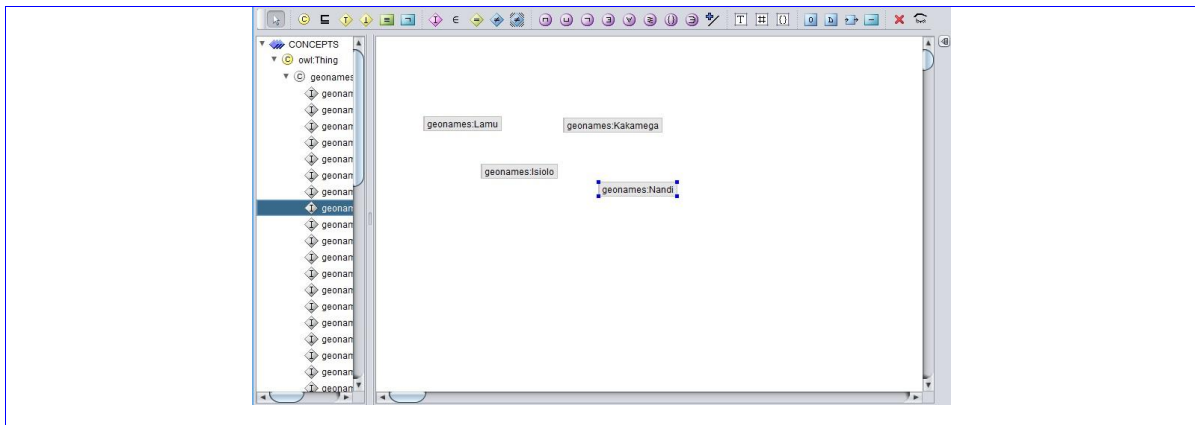
Requisitos no funcionales	No procede.
Asuntos pendientes	No procede.

Tabla 4: Descripción del Caso de uso Gestionar instancias.

Objetivo	El objetivo de la gestión de instancias es añadir, modificar y eliminar instancias de la ontología.	
Actores	Usuario.	
Resumen	El CU inicia cuando el usuario selecciona una instancia de la ontología. El CU termina cuando se inserta, modifica o elimina una instancia.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Abrir el editor de ontologías.	
Poscondiciones	Se inserta, modifica o elimina una instancia de la ontología exitosamente.	
Flujo de eventos		
Flujo básico Gestionar instancias		
	Actor	Sistema
1.	Selecciona la opción de abrir el editor de ontologías.	
2.		Muestra una interfaz que permite realizar las siguientes opciones: - Insertar una instancia. - Modificar una instancia. Ver Sección 1: "Modificar instancia". - Eliminar una instancia. Ver Sección 2: "Eliminar instancia".
3.	Selecciona la instancia a insertar.	
4.	Arrastra la instancia al panel principal.	
5.		Muestra las instancias en el panel principal. Termina el CU.
Prototipo de interfaz		

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.



Sección 1: “Modificar instancia”

Flujo básico Modificar instancia

	Actor	Sistema
2.	Inserta una nueva instancia.	
3.	Selecciona la instancia.	
4.	Elige la opción modificar.	
5.		Permite la modificación del nombre de la instancia.
6.	Escribe el nombre deseado.	
7.		Actualiza el nombre y el listado de instancias. Termina el CU.

Sección 2: “Eliminar instancia”

Flujo básico Eliminar instancia

	Actor	Sistema
2.	Selecciona una instancia del panel principal.	
3.		Marca la instancia seleccionada.
4.	Elige la opción eliminar.	
5.		Elimina la instancia. Termina el CU.

Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

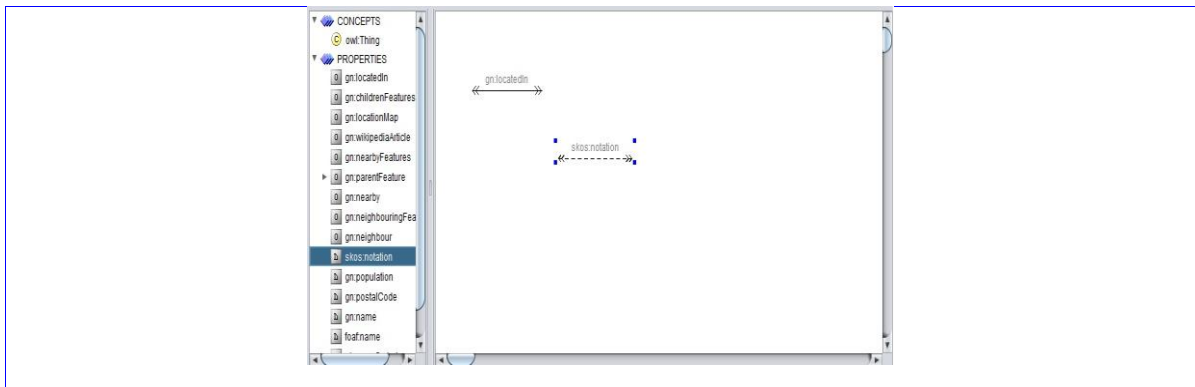
Requisitos no funcionales	No procede.
Asuntos pendientes	No procede.

Tabla 5: Tabla 4: Descripción del Caso de uso Gestionar propiedades.

Objetivo	El objetivo de la gestión de propiedades es añadir, modificar y eliminar propiedades de la ontología.	
Actores	Usuario.	
Resumen	El CU inicia cuando el usuario selecciona una propiedad de la ontología. El CU termina cuando se inserta, modifica o elimina una propiedad.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Abrir el editor de ontologías.	
Poscondiciones	Se inserta, modifica o elimina una propiedad de la ontología exitosamente.	
Flujo de eventos		
Flujo básico Gestionar propiedades		
	Actor	Sistema
1.	Selecciona la opción de abrir el editor de ontologías.	
2.		Muestra una interfaz que permite realizar las siguientes opciones: - Insertar una propiedad. - Modificar una propiedad. Ver Sección 1: "Modificar propiedad". - Eliminar una propiedad. Ver Sección 2: "Eliminar propiedad".
3.	Selecciona la propiedad a insertar.	
4.	Arrastra la propiedad al panel principal.	
5.		Muestra las propiedades en el panel principal. Termina el CU.
Prototipo de interfaz		

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.



Sección 1: “Modificar propiedad”

Flujo básico Modificar propiedad

	Actor	Sistema
2.	Inserta una nueva propiedad.	
3.	Selecciona la propiedad.	
4.	Elige la opción modificar.	
5.		Permite la modificación del nombre de la propiedad.
6.	Escribe el nombre deseado.	
7.		Actualiza el nombre y el listado de propiedades. Termina el CU.

Sección 2: “Eliminar propiedad”

Flujo básico Eliminar propiedad

	Actor	Sistema
2.	Selecciona una propiedad del panel principal.	
3.		Marca la propiedad seleccionada.
4.	Elige la opción eliminar.	
5.		Elimina la propiedad. Termina el CU.

Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

Requisitos no funcionales	No procede.
Asuntos pendientes	No procede.

Tabla 6: Descripción del Caso de uso Guardar ontología.

Objetivo	El objetivo es guardar la ontología creada.	
Actores	Usuario.	
Resumen	El CU inicia cuando el usuario selecciona la opción guardar. El CU termina cuando se guarda una ontología.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Abrir el editor de ontologías.	
Poscondiciones	Se guarda la ontología exitosamente.	
Flujo de eventos		
Flujo básico Gestionar propiedades		
	Actor	Sistema
1.	Selecciona la opción de abrir el editor de ontologías.	
2.		Muestra una interfaz que permite guardar una ontología.
3.	Selecciona la ontología a guardar.	
4.	Escoge la opción guardar.	
5.		Almacena la ontología. Termina el CU.
Prototipo de interfaz		

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

2.5.1 Arquitectura orientada a objetos

Las características de la arquitectura orientada a objetos son herencia, polimorfismo y el encapsulamiento; por tanto el desarrollo de sus componentes diseñados bajo este estilo de arquitectura son guiados por estos principios. En el diseño de la arquitectura del software, la interacción entre los objetos en las fases de modelado, diseño e implementación son el eje fundamental (Reynoso, y otros, 2004). El modelado del sistema será bajo este patrón arquitectónico, ya que el framework CARTOWEB, el cual es utilizado por GENESIG, posee una arquitectura orientada a objetos.

2.5.2 Arquitectura basada en componentes

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Se enfoca principalmente en descomponer el diseño en funcionalidades, a través de interfaces, lograr una comunicación con el sistema bien definida. La ventaja de ello es que provee un nivel de abstracción mayor y no se enfoca en los contenidos de los objetos como los protocolos de comunicación y la forma como se comparte el estado (Reynoso, y otros, 2004).

El patrón de arquitectura basado en componentes posee cuatro etapas: la etapa de análisis y comparación del proceso de desarrollo de software, análisis de arquitectura, la identificación de componentes y por último la especificación de los principales componentes a utilizar.

Es un estilo de diseño para aplicaciones compuestas de componentes individuales, los cuales ponen énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen las interfaces bien definidas. Igualmente, definen una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades. . El modelado del sistema también será bajo este patrón arquitectónico, ya que el framework CARTOWEB, posee una arquitectura basada en componentes.

2.6 Patrones de diseño

Los patrones de diseño son soluciones a problemas de desarrollo de software representados a través de un esquema y brindan una solución documentada a problemas de desarrollo de software Los patrones de diseño se dividen en dos grupos: GRASP Y GoF.

2.6.1 Patrones GRASP

Los patrones **General Responsibility Assignment Software Patterns** (GRASP, por sus siglas en inglés) son patrones generales de software para asignación de responsabilidades. Indican el nivel de

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

responsabilidad de la creación de un objeto o la implementación de un método de manera que se conceda toda la información necesaria para crear el sistema.

Experto: El objetivo de éste patrón es asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón se ve activo en la clase Ontología ya que posee información crea objetos, y delega operaciones de su clase hija Ontología_GeoNames.

Creador: Permite asignarle a la clase A la responsabilidad de crear una instancia de alguna clase guiando la asignación de responsabilidades relacionadas con la creación de objetos. El patrón se ve reflejado en las clases GeoName_Service, ya que crea un objeto de la clase GeoName_Source.

Alta Cohesión: El propósito de éste patrón es asignar a las clases responsabilidades de forma tal que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad, es decir que la cohesión sea alta. Esto significa que las clases con responsabilidades con estrecha relación no realicen gran trabajo con las clases conectadas. La acción de este patrón se ve sobre las clases GeoName_Source, Ontología, Ontología_GeoNames y Tesis_integral.

Bajo Acoplamiento: El bajo acoplamiento se basa en asignar las responsabilidades de forma tal que las clases se comuniquen con el menor número de clases que sea posible. Especifica cómo dar soporte a una dependencia insuficiente. El bajo acoplamiento se ve manifestado en todas las clases, ya que permitiendo así la independencia entre ellas.

Controlador: Expresa cómo asignar la responsabilidad del manejo de mensajes de los eventos del sistema a una clase, en otras palabras, controla el flujo de eventos del sistema y se lo asigna a clases específicas. Se muestra la operación de este patrón sobre la clase Ontología.

2.6.2 Patrones GoF

Además del uso de los patrones GRASP se tuvo en cuenta los patrones **GOF** acrónimo de **Gang Of Four**, los cuales se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Pressman, 2010).

Instancia única (Singleton): En el diseño de clases es necesario aplicar la solución del patrón Singleton que garantiza el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. En la clase *ServerContext* se ve reflejada la acción de este patrón, ya que crean una instancia única del mapa con el método *MapObj*.

Acción (Command): Se utiliza en el proceso de petición mediante *Graphic User Interface* (GUI, por sus siglas en inglés) al sistema de una información cualquiera por un cliente. Uno de los aspectos más

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

importantes son las GUI, pues el usuario interactúa constantemente con ellas y por eso se aplica la solución propuesta por este patrón de diseño. En la clase *AJAXHelper* se da a conocer la acción del patrón *Command*, ya que es la clase que encapsula las peticiones del usuario y la encargada de relacionar las interfaces con el servidor.

2.7 Modelo de diseño

El diseño del software es un proceso de varios pasos que se clasifican dentro uno mismo. En general, la actividad del diseño se refiere al establecimiento de la estructura de datos, la arquitectura general del software y las representaciones de interfaz y algoritmos. El proceso de diseño traduce los requisitos en una representación de software (Pressman, 2010).

Con del modelo de diseño se crea un plano del modelo de implementación, a través de la captura de los requisitos individuales, las clases y las interfaces. Procura lograr ahorrar trabajo a la hora de implementación a través la manipulación de elementos relacionados con los requisitos funcionales y no funcionales.

2.7.1 Diagrama de clases del diseño

Un diagrama de clases del diseño es un artefacto en el que se representan los conceptos en un dominio del problema. Representa además, el funcionamiento del mundo real, no de la implementación automatizada del mismo (Pressman, 2010).

En el diagrama de clases del diseño (figura 5) se expone la estructura que presenta CARTOWEB. A continuación se realiza una breve descripción de las clases representadas:

MapObj: Es donde se definen todas las funciones y métodos para intercambiar datos con el Servidor de Mapas.

PluginManager: Es la clase donde se realiza la gestión de todos los plugins existentes.

ServerContext: Contiene la información común que debe ser utilizada por el cliente y el servidor, utilizando la información escogida como un objeto para efectuar un cómodo manejo de datos.

CartoClient: Administra la información de los datos y las funciones realizadas por *formOntweb.ui.js* y *formOntweb.js*, definiendo así un conjunto de variables globales preparadas para ser utilizadas por la aplicación.

Client: Contiene todos los archivos de PHP del lado de *CartoClient* y concede el permiso de interactuar *Index.php* con *CartoClient*.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

ServerPlugin: Suministra una base de herramientas necesarias para desarrollar plugins.

ClientPlugin: Destinada a servir del lado del cliente. Contiene las interfaces para los plugins del lado del cliente.

ServerOntweb: Es donde se encuentran los métodos de llamada a los servicios web. Es el encargado de contener las llamadas a toda la información solicitada por el cliente.

ClientOntweb: Se encarga de realizar las peticiones al servidor para obtener la información necesaria.

RequestOntweb: Manipula las llamadas realizadas por el ClientOntweb.

ResultOntweb: Manipula las respuestas de las llamadas realizadas por el cliente.

CWSerializable: Es la encargada de serializar todas las clases, con el objetivo de transferir datos a través del protocolo SOAP, permitiendo el intercambio de información entre ClientOntweb y ServerOntweb.

AJAXHelper: Su principal objetivo es enviar las respuestas (request) de los plugins AJAX, para nutrir al plugin o los plugins que responden a las peticiones del usuario.

AJAXOntweb: Se encarga de gestionar el request y response de las peticiones del usuario por AJAX.

formOntweb.iu.js: Contiene el código de la visualización de la aplicación, paneles, ventanas, botones, grids, formularios, entre otros.

formOntweb.js: Es la encargada de implementar los métodos que serán llamados desde formOntweb.iu.js, para ejecutar acciones y métodos en el ServerOntweb.

Index.php: Recibe las peticiones realizadas por el cliente, las gestiona y ejecuta la función de ordenar a construir el ClientPage.

Ontología: Realiza la función de integrar una ontología del dominio geográfico con la Ontología_GeoNames.

Ontología_GeoNames: Su función radica en crear conceptos, instancias, propiedades de datos y de objetos a partir de la información extraída por la clase GeoNames_Source.

GeoNames_Source: Se ocupa de conectarse al sitio de GeoNames, y consumir los servicios web que brinda.

Tesis_ultima_version: Clase Main (Principal).

Servicio: Cumple la función de publicar las WSDLs de los servicios web para ser llamadas por el ServerOntweb.php.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

UnionOnto: Clase contenedora del servicio web *union*.

GenomeService: Clase que contiene el servicio web *getListXLocation*.

GeoName_Children: Clase contenedora del servicio web *getListXchildren*.

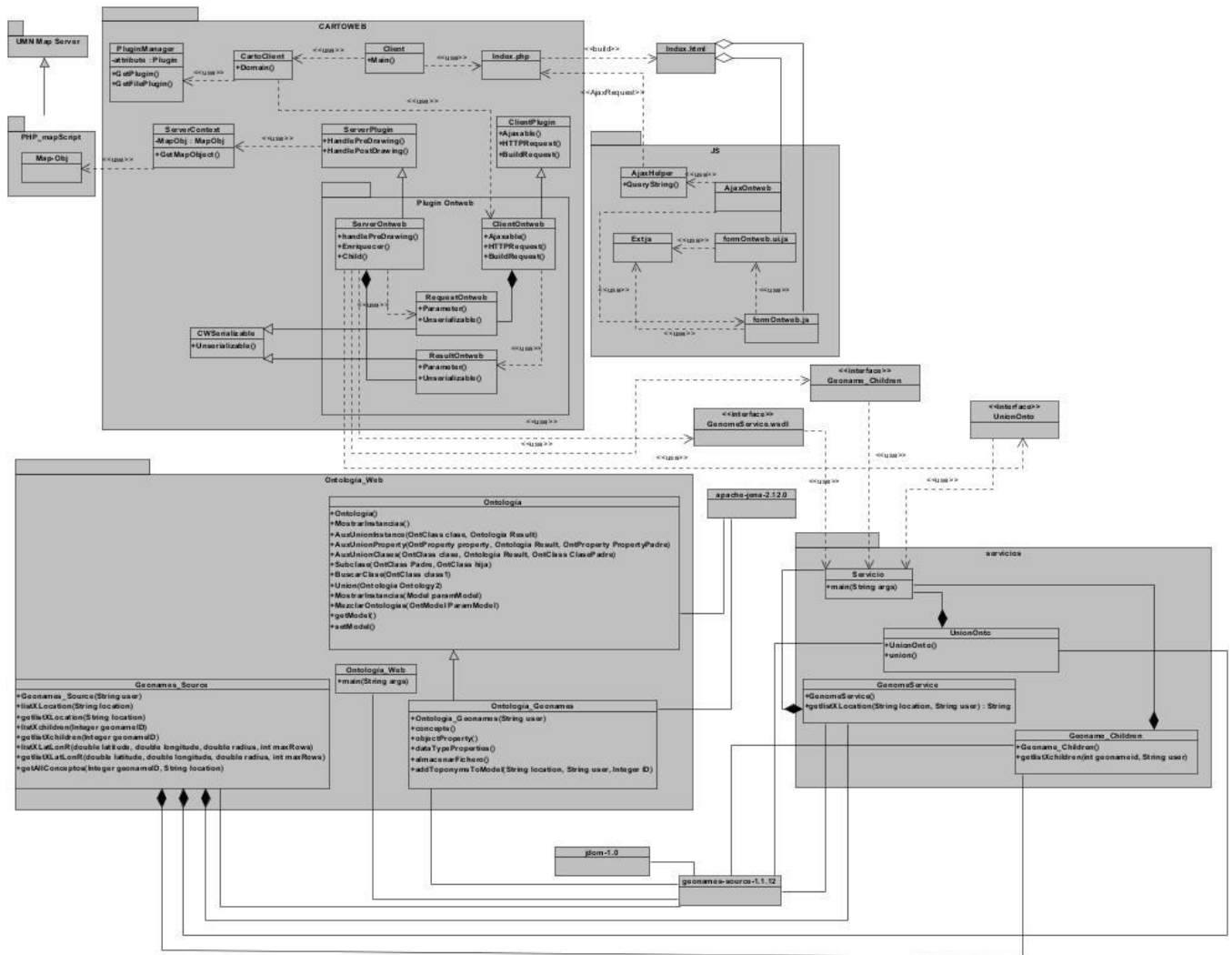


Figura 5: Diagrama de clases del diseño del Caso de Uso Enriquecer ontología.

Conclusiones del capítulo

Al concluir este capítulo se permitió conocer las características, la arquitectura y el diseño de la plataforma GENESIG, además ideó la propuesta de solución del sistema. Se dio paso a la confección de los diagramas de dominio, de casos de uso y una explicación del modelo de negocio, para lograr un mejor entendimiento de la herramienta de software a desarrollar y se forjaron las bases de su creación. Se diseñó la especificación de los requisitos, tanto funcionales como no funcionales y la descripción de

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPITULO 2: TENDENCIAS Y SOPORTE TECNOLÓGICO.

los actores del sistema para forjar el camino de las funcionalidades que debe cumplir el sistema para satisfacer las necesidades.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Introducción

En el presente capítulo se especifica la arquitectura de software orientada a objetos y la arquitectura basada en componentes, ambas utilizadas como base para la construcción del software. Se hace referencia a los patrones de diseño manipulados por las diversas clases del esquema de software. En el diseño de software se expone el diagrama de clases del diseño para lograr tener un plano del modelo de implementación.

3.1 Modelo de implementación

La implementación del sistema comienza con la creación de los componentes, *scripts*, ficheros de código fuente, de código binario, ejecutables, entre otros. Está compuesto por una serie de componentes y funcionalidades del sistema que en conjunto logran su correcto funcionamiento. A través del mismo se obtienen tanto el diagrama de componentes como el diagrama de despliegue.

3.1.1 Diagrama de componentes

Los diagramas de componentes detallan físicamente los elementos del sistema y sus relaciones. Asimismo, muestra la relación entre los componentes de software, sus dependencias, comunicaciones, condiciones entre otras condiciones.

A continuación en la figura 6, se muestra el diagrama de componentes del sistema:

CAPÍTULO 3: Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.

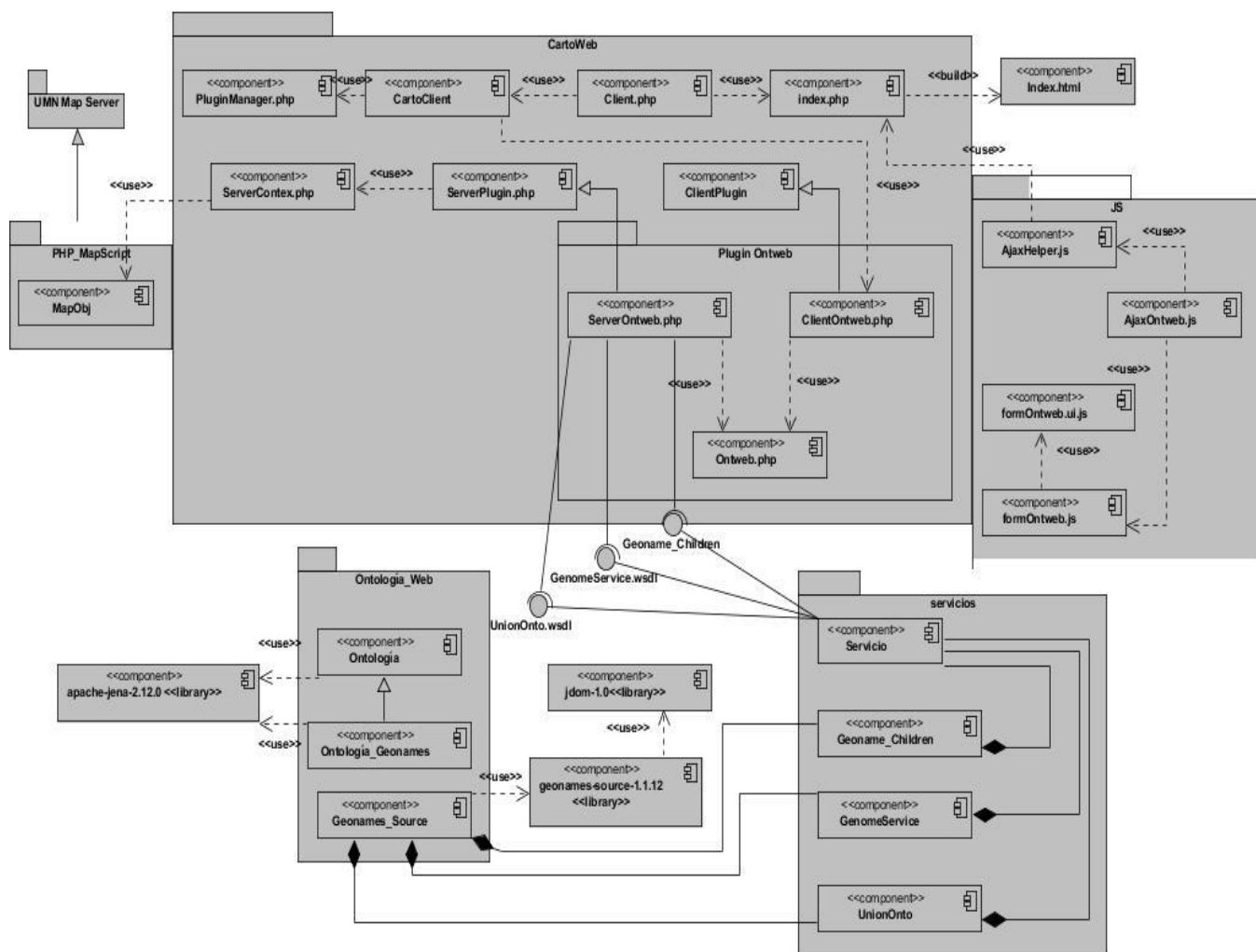


Figura 6: Diagrama de componentes del Caso de Uso Enriquecer ontología.

3.1.2 Diagrama de Despliegue

El diagrama de despliegue muestra las relaciones físicas entre los componentes (hardware y software) del sistema. Es un conjunto de nodos enlazados por conexiones de comunicación. El objetivo principal no es realizar una descripción de la infraestructura, sino describir el camino que deben seguir los componentes para corresponder a la aplicación que es desplegada a través del mismo.

En la figura 7 se muestra el diagrama de despliegue del sistema propuesto, en el cual se da a conocer la estructura de los componentes. Consta de un cliente que se conectará y realizara consultas a través de un protocolo *HTTP* al Servidor Web de GENESIG, y este servidor es el encargado de hacer las peticiones al Servidor de servicios de ontologías utilizando el protocolo *SOAP*.

CAPÍTULO 3: Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.

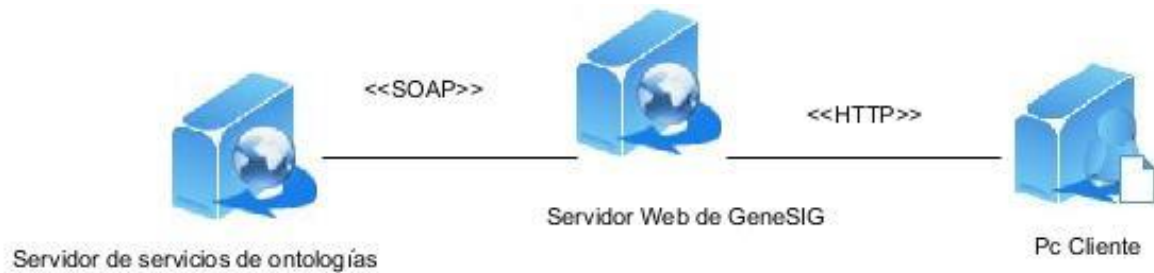


Figura 7: Diagrama de despliegue.

3.2 Pruebas del sistema propuesto

A todo sistema se le deben realizar las pertinentes pruebas de software para conocer la calidad, el nivel de optimización y rendimiento del software.

3.2.1 Pruebas de caja negra

Las pruebas de caja negra son las que se llevan a cabo sobre la interfaz del software. Con la realización de los casos de prueba se procura demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y que se produce un correcto resultado. Los principales objetivos que se persiguen al realizar este tipo de prueba son:

- Detectar el incompleto o incorrecto funcionamiento del software.
- Detectar errores de interfaz.
- Errores de iniciación o terminación.
- Problemas de rendimiento.

O sea, en otras palabras se pretende demostrar que las funcionalidades del software son operativas, que se tienen de respuestas resultados correctos para las pertinentes entradas.

Así mismo, existen un conjunto de técnicas de pruebas las cuales facilitarán que se compruebe el grado de cumplimiento de los requisitos funcionales, entre ellas se encuentran:

- Partición de equivalencia.
- Análisis de valores límite.
- Análisis causa-efecto.
- Métodos basados en grafos.
- Pruebas de comparación.

Se empleó la técnica de prueba Partición equivalente, por ser una de las más efectivas, ya que permite conocer los valores válidos (V), inválidos (I) y dato irrelevante por el sistema (N/A). La partición de

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 3: Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.

equivalencia se enfoca en desarrollar casos de prueba que descubran clases de errores, reduciendo el número de casos de prueba a desarrollar. A continuación se muestran los resultados de la prueba realizada al caso de uso Enriquecer ontología.

Tabla 7: Diseño de Casos de Prueba para el caso de uso Enriquecer ontología.

Escenario		Descripción	Continentes	País	Filtrar	Respuesta del sistema	Flujo central
SC1: Enriquecer ontología.	EC 1.1 Enriquecer ontología correctamente.	Se crea la ontología. El sistema valida que se hayan seleccionado los datos para su construcción.	V	V	V	Muestra los datos del continente, país y/o filtro. Muestra un mensaje indicando que se creó y guardó correctamente la ontología.	Cuando se selecciona un continente, el sistema muestra la opción para seleccionar un país y/o filtrar por el nombre de un lugar. Asimismo, puede obtener respuesta de su consulta con solo filtrar por nombre de lugar. Después solo queda seleccionar la opción de crear ontología.
			Asia	Japón	Tokio		
			V	V	V		
			Asia	Japón	vacío		
			V	V	V		
Asia	vacío	vacío					
SC1: Enriquecer ontología.	EC 1.1 Enriquecer ontología correctamente.	Se crea la ontología. El sistema valida que se hayan seleccionado los datos para su construcción.	V	V	V	Muestra los datos del continente, país y/o filtro. Muestra un mensaje indicando que se creó y guardó correctamente la ontología.	Cuando se selecciona un continente, el sistema muestra la opción para seleccionar un país y/o filtrar por el nombre de un lugar. Asimismo, puede obtener respuesta de su consulta con solo filtrar por nombre de lugar. Después solo queda seleccionar la opción de crear ontología.
			Asia	Japón	Tokio		
			V	V	V		
			Asia	Japón	vacío		
			V	V	V		
Asia	vacío	vacío					
SC1: Enriquecer ontología.	EC 1.1 Enriquecer ontología correctamente.	Se crea la ontología. El sistema valida que se hayan seleccionado los datos para su construcción.	V	V	V	Muestra los datos del continente, país y/o filtro. Muestra un mensaje indicando que se creó y guardó correctamente la ontología.	Cuando se selecciona un continente, el sistema muestra la opción para seleccionar un país y/o filtrar por el nombre de un lugar. Asimismo, puede obtener respuesta de su consulta con solo filtrar por nombre de lugar. Después solo queda seleccionar la opción de crear ontología.
			Asia	Japón	Tokio		
			V	V	V		
			Asia	Japón	vacío		
			V	V	V		
Asia	vacío	vacío					
SC1: Enriquecer ontología.	EC 1.1 Enriquecer ontología correctamente.	Se crea la ontología. El sistema valida que se hayan seleccionado los datos para su construcción.	V	V	V	Muestra los datos del continente, país y/o filtro. Muestra un mensaje indicando que se creó y guardó correctamente la ontología.	Cuando se selecciona un continente, el sistema muestra la opción para seleccionar un país y/o filtrar por el nombre de un lugar. Asimismo, puede obtener respuesta de su consulta con solo filtrar por nombre de lugar. Después solo queda seleccionar la opción de crear ontología.
			Asia	Japón	Tokio		
			V	V	V		
			Asia	Japón	vacío		
			V	V	V		
Asia	vacío	vacío					
SC1: Enriquecer ontología.	EC 1.2 Enriquecer ontología incorrectamente.	El usuario deja campos vacíos	I vacío	N/A	I vacío	Muestra mensaje de error informando al usuario sobre el/los error(es) cometido(s).	Cuando se selecciona el campo de filtrar pero no se especifica ningún criterio de búsqueda o no se selecciona ningún continente.

Se realizaron 3 iteraciones de las pruebas de caja negra, las cuales dieron como resultados:

En la primera iteración se obtuvo como resultado un total de 6 no conformidades de ellas 4 significativas relacionadas con errores de funcionalidad de la aplicación, las 2 restantes fueron no significativas, asociadas a errores ortográficos y de redacción.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

CAPÍTULO 3: Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.

En la segunda iteración se encontraron un total de 3, siendo 2 significativas asociadas a problemas con las funcionalidades del software y 1 no significativa perteneciente a errores ortográficos.

En la tercera iteración no se encontraron no conformidades, siendo satisfactorio el resultado de las pruebas realizadas al software.

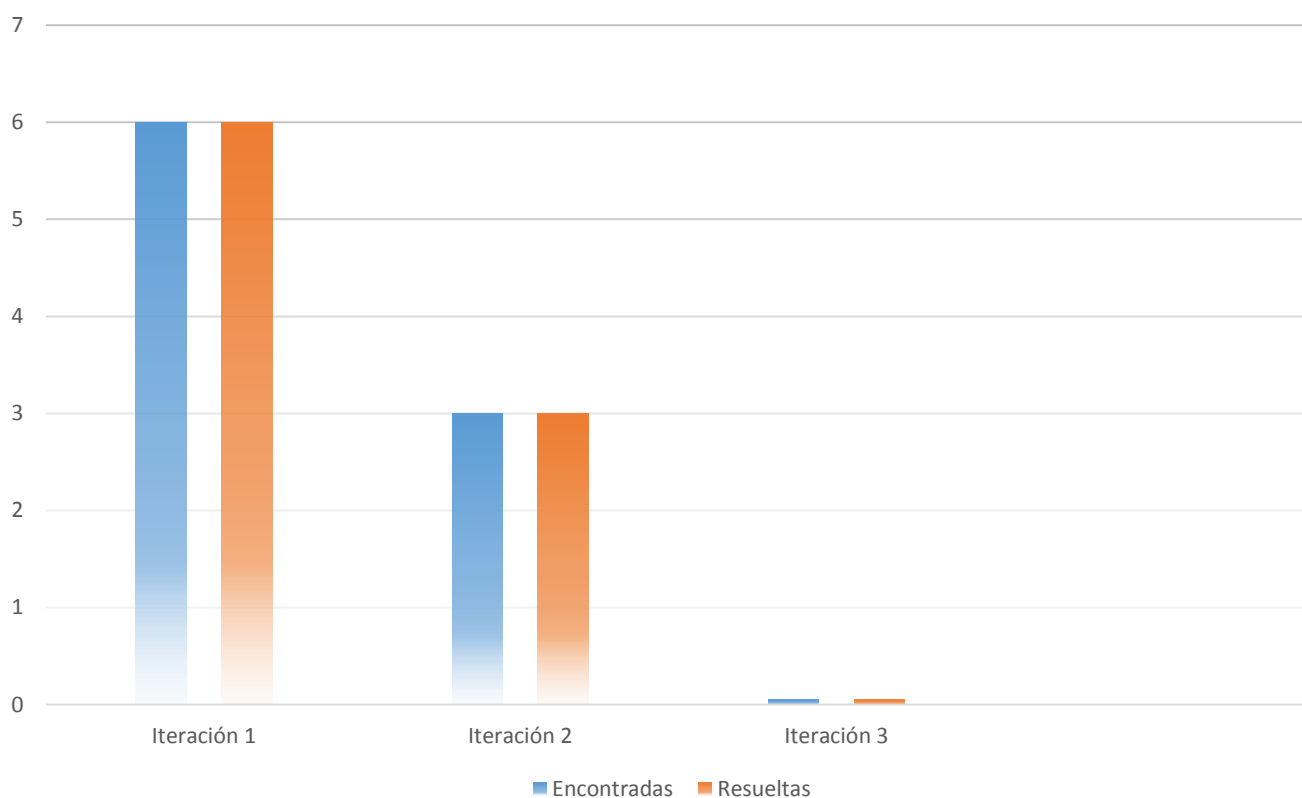


Figura 8: Resultados de las pruebas aplicadas.

Conclusiones del capítulo

A lo largo de este capítulo, se trazó una guía de despliegue del software a través de los patrones arquitectónicos, así como el asentamiento de las bases para la implementación con los diagramas de diseño y despliegue respectivamente. Se asignaron las pertinentes responsabilidades a las clases creadas en el modelo de diseño a través de los patrones de diseño, permitiendo una mejor organización y optimización del código. Una vez terminado el proceso de implementación y despliegue se ejecutaron las pruebas al sistema, aplicando la prueba de caja negra con el objetivo de detectar y evaluar el correcto cumplimiento de la interfaz.

CONCLUSIONES

Una vez concluida la presente investigación, se tiene como resultado una herramienta de software que permite enriquecer una ontología geográfica, por medios semiautomáticos, a partir del análisis de GeoNames. Se cumplen los objetivos propuestos para este trabajo en función de los objetivos trazados para llevar a cabo el proceso investigativo.

Por otra parte se arriban a las siguientes conclusiones:

Los objetivos específicos trazados facilitaron la organización de la investigación, así como un mejor entendimiento del objetivo general.

Se capturaron los requisitos funcionales y no funcionales con los cuales debe cumplir el sistema, lo cual forjó las bases para la construcción y el perfeccionamiento del trabajo, cumpliendo así los objetivos deseados.

Se logró la realización del diseño, construcción e implementación del sistema teniendo en cuenta los patrones arquitectónicos basados en componentes, orientado a objetos, así como su combinación con los patrones de diseño.

Se obtuvo un plugin (Ontweb) que cumple con los requisitos y las necesidades definidas.

Las pruebas de caja negra las cuales arrojaron resultados satisfactorios, comprobaron el funcionamiento del sistema, demostrando la fiabilidad del software desarrollado.

RECOMENDACIONES

1. Aumentar el enriquecimiento de la ontología geográfica, a partir de la inclusión de la generación automática de relaciones topológicas, teniendo en cuenta la ubicación espacial de los objetos.
2. Inclusión de otras fuentes de información geográficas disponibles en internet.

REFERENCIAS BIBLIOGRÁFICAS

Achour, M. y F. Betz. 2006. PHP. *PHP Manual*. [En línea] 2006. <http://php.net>.

Balani, N. y Hathi, R. 2009. *Apache CXF Web Service Development*. ISBN . s.l. : Edtion ed: Packt Publishing, 2009. 978-1-847195-40-1.

Barrera, Milagros, Nuñez, Haydemar y Ramos, Esmeralda. 2012. *Ingeniería Ontológica*. Caracas : s.n., 2012.

Buscaldi, Davide, Rosso, Paolo y García, Piedachu Peris. 2006. *Inferring Geographical Ontologies from Multiple Resources for Geographical Information*. Universidad Politécnica de Valencia. 2006. pág. 4.

CEGIS. 2015. Center of Excellence for Geospatial Information Science. *Center of Excellence for Geospatial Information Science*. [En línea] 2015. <http://cegis.usgs.gov/index>.

Cordero, Jorge Luis. 2011. *Metodologías Ágiles " Proceso Unificado Ágil (AUP)"*. La Paz, EL Alto : s.n., 2011.

Díaz, Orestes Febles. 2012. *MIDAC: MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS BASADAS EN ARQUITECTURAS ORIENTADAS A SERVICIOS*. Universidad de las Ciencias Informáticas. La habana : s.n., 2012. pág. 21, Tesis doctoral.

Ehrig, Marc. 2007. *Ontology Alignment: Bridging the Semantic Gap*. s.l. : Springer, 2007.

Euzenat, J y and Shvaiko, P. 2007. *Ontology Matching*. s.l. : Berlin - Heidelberg - New York: Springer, 2007.

Fu, G., Jones, C.B. y Abdelmoty, A.I. 2005. *Building a Geographical Ontology for Intelligent Spatial Search on the Web*. Innsbruck, Austria. : s.n., 2005. IASTED International Conference on Databases and Applications, part of the 23rd Multi-Conference on Applied Informatics.

García, María de la Nava Maroto. 2007. *Las relaciones conceptuales en la terminología de los productos cerámicos y su formalización mediante un editor de ontologías*. 2007. Tesis doctoral.

Garea Llano, Eduardo. 2007. *Estado actual de la interpretación semántica de datos espaciales"*. Centro de Aplicaciones de Tecnología de Avanzada. Habana. : s.n., 2007.

GeoNames. 2015. GeoNames. *GeoNames*. [En línea] 2015. <http://www.geonames.org/>.

Gruber, Thomas Robert. 1993. *Towards Principles for the Design of Ontologies Used for Knowledge Sharing*. Deventer, The Netherlands. : Kluwer Academic Publishers, 1993. págs. 907-928.

Guarino, Nicola. 1998. *Formal Ontology and Information Systems in Proceeding of FOIS '98*. Trento, Italy : s.n., 1998.

Guarino, Nicola. 1997. *Understanding, building and using ontologies*. 1997. págs. 46(2-3).p. 293-310.

Hess, Guillermo Nudelman. 2008. *Towards Effective Geographic Ontology Semantic Similarity Assessment*. Programa de Pós-Graduação em Computação., UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Porto Alegre. : s.n., 2008. pág. 118, Tesis doctoral.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

- Jones, Christopher y Abdelmoty, Alia. 2002.** SPIRIT. *Spatially-Aware Information Retrieval on the Internet*. [En línea] 2002. <http://www.geo-spirit.org/>.
- Larman, C. 1999.** *UML y Patrones: Introducción al análisis y programación orientada a objetos*. s.l. : Prentice Hall, Pearson Education, 1999. pág. 624. Vol. 2da Edición.
- Letelier, Patricio y Penadés, M^a Carmen. 2006.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming(XP)*. Departamento de Sistemas de Informáticos y Computación., Universidad Politécnica de Valencia. Valencia : s.n., 2006. pág. 17. 46022.
- Marcos, Antonio y González, Huerta. 2012.** Mantenimiento Dinámico de las aplicaciones desarrolladas con el marco de trabajo jWebSocket. [En línea] Universidad de las Ciencias Informáticas, 2012. http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_06163_12..
- Martínez, Jorge Iván Meza. 2008.** *Desarrollo de un EJB para la georreferenciación utilizando webservices y el servicio de GeoNames*. 2008.
- Martínez, Manuel Enrique Puebla y Águila, Adrián Gracia. 2012.** No.3, La habana : s.n., 2012, Vol. Vol.6, pág. 6. ISSN: 1994-1536 | e-ISSN: 2227-1899.
- Martínez, Manuel Enrique Puebla, Ortega, Jose M. Perea y Cuevas, Alfredo Simón. 2015.** *Integración de técnicas browsing con ontologías en un modelo de recuperación de información geográfica*. 2015. págs. 53-61.
- Mata Rivera, Miguel. Félix. 2009.** *Recuperación y Ponderación de Información Geográfica desde Repositorios no Estructurados Conducidas por Ontologías*. Instituto Politécnico Nacional. Mexico. : s.n., 2009. Tesis doctoral.
- McGuinness, Deborah L. y Van Harmelen, Frank. 2004.** World Wide Web Consortium. [En línea] 10 de febrero de 2004. <http://www.w3.org>.
- Murugarren, Joaquin Gracia. 2011.** WebEstilo. Manual de PHP. [En línea] 1 de junio de 2011. <http://www.webestilo.com/php/>.
- Navarro, Juan Francisco García. 2012.** *Analítica Visual aplicada a la Ingeniería de Ontologías*. Salamanca : s.n., 2012.
- Oracle, Glassfish Server. 2015.** *Oracle Glassfish Server*. 2015. Release Notes.
- Ping, D. y Yong, L. 2009..** *Building Place Name Ontology to Assist in Geographic Information Retrieval*. 2009. págs. 306–309. Presented at the Computer Science-Technology and Applications, IFCSTA '09. International Forum on, IEEE, Chongqing, China. .
- Pressman, Roger S. 2008.** *Ingeniería de software.Un enfoque práctico*. 5ta. Madrid : McGraw Hill, 2008. pág. 765.
- Pressman, Roger S. 2012.** *Métodos convencionales para la ingeniería del software*. 2012. pág. 94. Vols. Quinta Edición,Parte III.
- . 2010. *Software Engineering.A Practitioner's Approach*. New York : s.n., 2010. s.n..
- RAE. 2015.** Real Academia Española. *Real Academia Española*. [En línea] 2015. <http://www.rae.es>.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

- Ramos, E y Núñez, H. 2007.** *Ontologías: componentes, metodologías, lenguajes, herramientas y aplicaciones.* Caracas. : s.n., 2007. Lecturas en Ciencias.
- Reynoso, C. y Kicillof, N. 2004.** *Estilos y patrones en la estrategia de arquitectura de Microsoft.* 2004.
- Ruiz, Miguel Jesús Torres. 2007.** *Representación Ontológica basada en Descriptores Semánticos aplicada a Objetos Geográficos.* Centro de Investigación en Computación, Instituto Politecnico Nacional. Mexico : s.n., 2007. pág. 425, Tesis doctoral.
- Rumbaugh, Jacobson. 2000.** *El Lenguaje Unificado de Modelado. Manual de referencia.* Madrid : Pearson Educación, 2000.
- Schreiber, Guus y Raimond, Yves. 2014.** Word Wide Web Consortium. [En línea] W3C Working Group, 24 de June de 2014. <http://www.w3.org>.
- Sommerville, Ian. 2005.** *Ingeniería de Software.* [ed.] PEARSON ADDISON WESLEY. 7ma edición. Madrid : PEARSON EDUCACIÓN S.A, 2005. 84-7829-074-5.
- W3C. 2015.** Word Wide Web Consortium. *Word Wide Web Consortium.* [En línea] 2015. <http://www.w3.es>.
- Wikipedia. 2015.** Wikipedia. *Wikipedia.* [En línea] 2015. <http://es.wikipedia.org>.
- Yeh, J.-H. y Yang, N. 2008.** *Ontology Construction Based on Latent Topic Extraction in a Digital Library.* Springer-Verlag. Berlin, Heidelberg : s.n., 2008. págs. pp. 93–103. in: Proceedings of the 11th International Conference on Asian Digital Libraries: Universal and Ubiquitous Access to Information, ICADL 08.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

ANEXOS

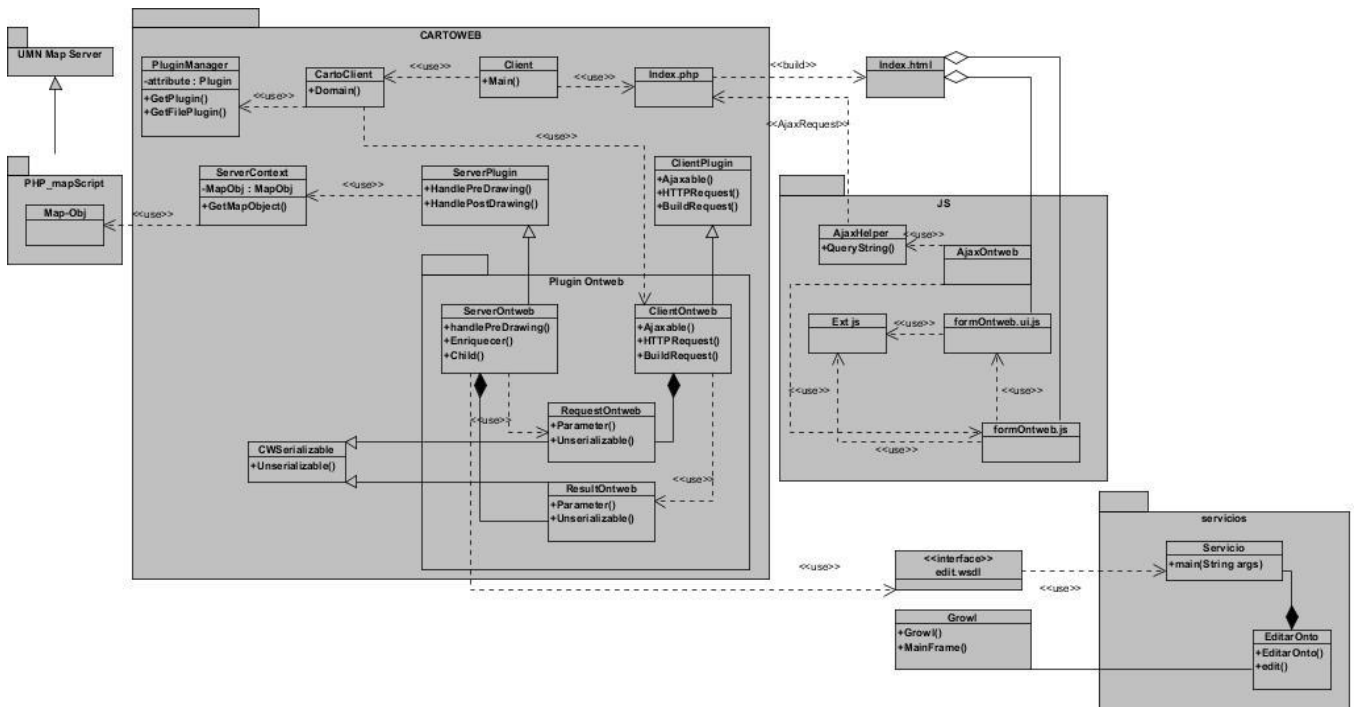


Figura 9: Diagrama de clases del diseño del Caso de Uso Gestionar conceptos.

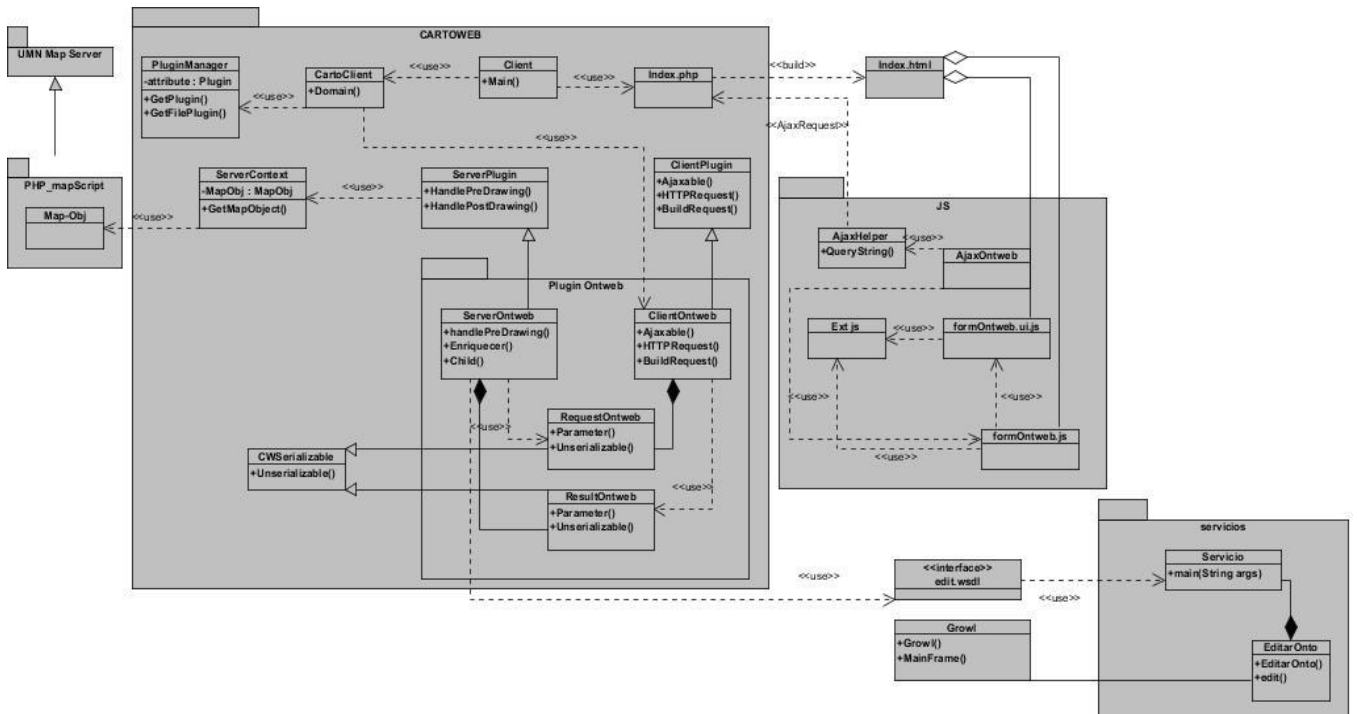


Figura 10: Diagrama de clases del diseño del Caso de Uso Gestionar instancias.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

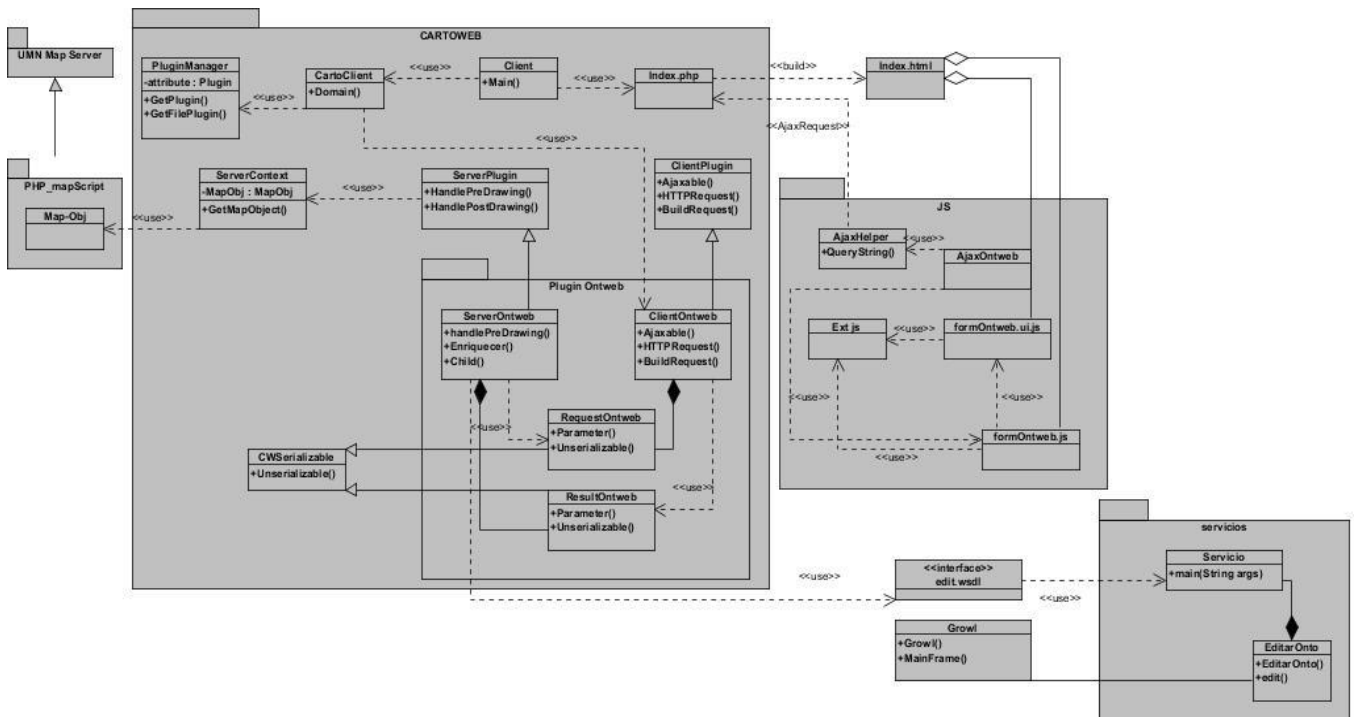


Figura 11: Diagrama de clases del diseño del Caso de Uso Gestionar propiedades.

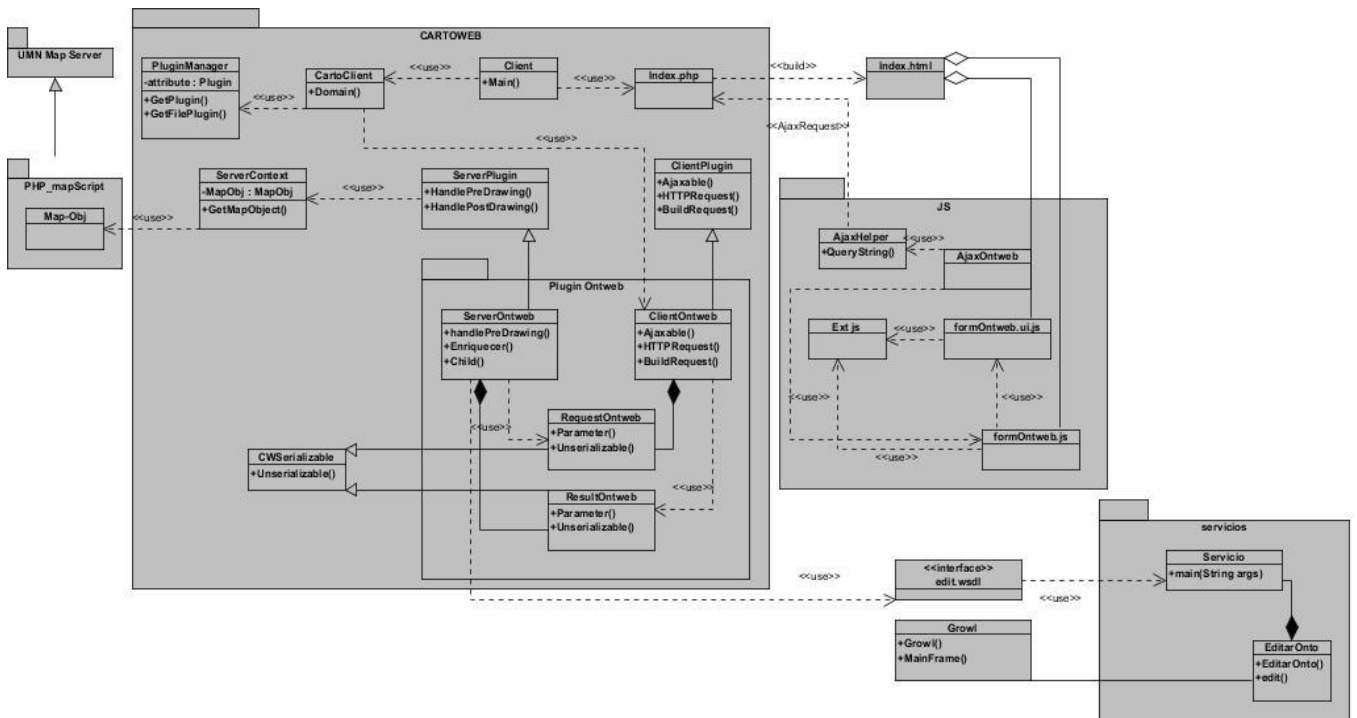


Figura 12: Diagrama de clases del diseño del Caso de Uso Guardar ontología.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir de fuentes disponibles en Internet.

Tabla 8: Diseño de Casos de Prueba para el caso de uso Guardar ontología.

Escenario	Descripción	Respuesta del sistema	
EC 1.1 Guardar ontología correctamente.	Guarda los componentes ontológicos correctamente.	Crea un nuevo fichero con el nombre y la dirección especificada.	Ir al menú archivo, la opción salvar, e indicar el nombre y la extensión del archivo a guardar. Por ultimo seleccionar "Guardar".
EC 1.2 Guardar ontología incorrectamente.	El usuario no guarda la ontología correctamente.	Muestra mensaje de error informando al usuario sobre el/los error(es) cometido(s).	El usuario selecciona la opción cancelar o guarda el archivo con una extensión incorrecta.

Tabla 9: Diseño de Casos de Prueba para el caso de uso Gestionar conceptos.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Gestionar conceptos correctamente.	Gestiona los conceptos correctamente.	Inserta, modifica, o elimina un concepto.	El usuario selecciona y arrastra los conceptos hacia el panel principal del editor de ontologías.
EC 1.2 Gestionar conceptos incorrectamente.	El usuario realiza mal la gestión de conceptos.	—	El usuario selecciona los conceptos pero no los arrastra hacia el panel principal.

Tabla 10: Diseño de Casos de Prueba para el caso de uso Gestionar instancias.

**Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos,
a partir de fuentes disponibles en Internet.**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Gestionar instancias correctamente.	Gestiona las instancias correctamente.	Inserta, modifica, o elimina una instancia.	El usuario selecciona y arrastra las instancias hacia el panel principal del editor de ontologías.
EC 1.2 Gestionar instancias incorrectamente.	El usuario realiza mal la gestión de instancias.	—	El usuario selecciona las instancias pero no las arrastra hacia el panel principal.

Tabla 11: Diseño de Casos de Prueba para el caso de uso Gestionar propiedades.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Gestionar propiedades correctamente.	Gestiona las propiedades correctamente.	Inserta, modifica, o elimina una propiedad.	El usuario selecciona y arrastra las propiedades hacia el panel principal del editor de ontologías.
EC 1.2 Gestionar propiedades incorrectamente.	El usuario realiza mal la gestión de propiedades.	—	El usuario selecciona las propiedades pero no las arrastra hacia el panel principal.