



Universidad de las Ciencias Informáticas

Facultad 6

Trabajo de diploma para optar por el título

Ingeniero en Ciencias Informáticas

Título: Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Autores:

Jorge Luis Espinosa Guerrero.

Yasiel Rodríguez Hernández.

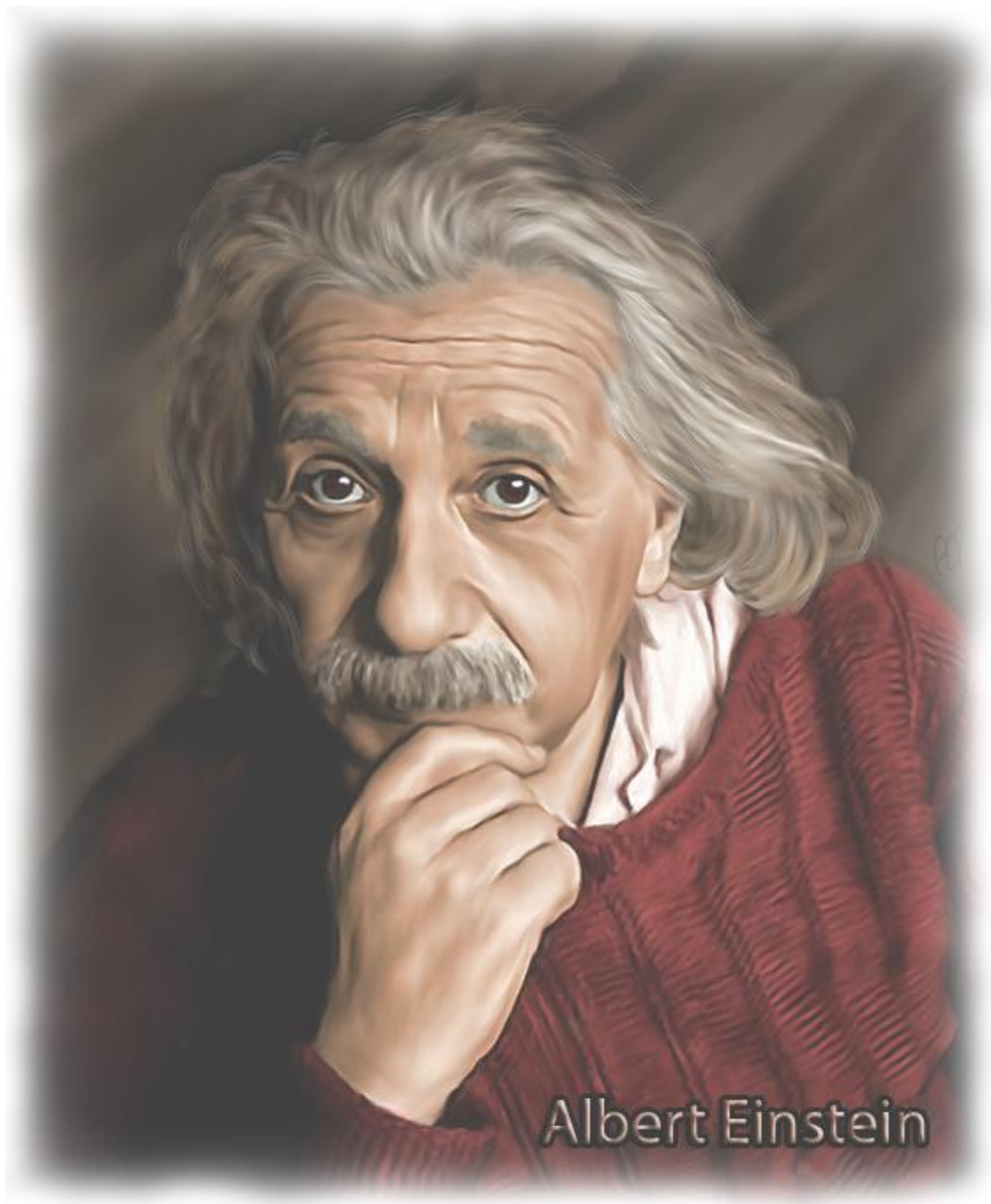
Tutor: Msc. Manuel Enrique Puebla Martínez.

La Habana, Junio del 2015

“Año 57 de la Revolución”

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”



DECLARACIÓN DE AUTORÍA

Declaro ser el único autor de la presente tesis 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 a los ____ días del mes de _____ del año 2015.

Firma del Autor

Jorge Luis Espinosa Guerrero

Firma del Autor

Yasiel Rodríguez Hernández

Firma del Tutor

Msc. Manuel Enrique Puebla
Martínez

DEDICATORIA

A mis padres, por ser mi inspiración para llegar hasta aquí.

A mis abuelos Rigo y Rubi, por ser tan especiales para mí.

Jorge Luis

AGRADECIMIENTOS

A toda mi familia, por apoyarme y confiar en mí en todos los momentos difíciles en estos 5 años en la UCI, especialmente a mis padres y mi tío José Ramón; las personas que más quiero, respeto y admiro en la vida.

A todos mis amigos de batallas Yaikel, Yobi, Dairelis, y especialmente el Yoa.

A mis amigas Jessy y Glennis, que son como hermanas; por todo lo que vivimos y compartimos en cada uno de estos años.

Armando y Yulina, gracias por tratarme como su hijo, y por desempeñar el papel de padres dedicados, atentos y preocupados por mi futuro en cada momento.

A mi amiga Anita Espinosa, por ayudarme cuando más lo necesitaba, al igual que Enier, Luis Enrique, Rosayne, Pedro y Dashiel.

Y por último pero no menos importante, a mi novia Katy, por entenderme, apoyarme, preocuparse y soportarme en mis días de estrés en los que trabaja en la tesis.

Para no dejar nadie afuera, quiero terminar agradeciendo a todas las personas que de alguna manera me ayudaron a lo largo de mi carrera, para poder estar parado aquí frente a mis padres y amigos.

Jorge Luis

RESUMEN

La comunidad científica concuerda en que el uso de las ontologías es fundamental para el procesamiento de la semántica, en particular la espacial. Para enriquecer una ontología se necesita la incorporación de los componentes ontológicos con que cuenta la misma; hacerlo de manera manual implica la utilización de un editor de ontologías además de tener conocimiento sobre ingeniería ontológica para lograr la edición. Esto implica tiempo y esfuerzo en dicha tarea, y como seres humanos, existe la posibilidad de cometer errores. Esta investigación propone un método para el desarrollo de una herramienta de software que me permita disminuir el costo en tiempo y esfuerzo del enriquecimiento de ontologías geográficas de gran tamaño, a partir del análisis de base de datos espaciales y de manera semiautomática. Con la extracción de la información contenida en la base de datos, se obtienen los componentes ontológicos, que posteriormente permitirá la construcción de una ontología con el uso del framework Jena. Luego se integra la misma con la ontología del dominio geográfico *Kaab-Ontology*, incorporándole nuevos componentes a la misma. La ontología resultante de la integración, se visualiza en la plataforma GeneSIG donde el usuario final puede interactuar con ella y editarla. El desarrollo del método permitió disminuir el costo en esfuerzo y tiempo del proceso de enriquecimiento de una ontología del dominio geográfico.

Palabras claves: base de datos espaciales, enriquecimiento semántico, ontología, semántica espacial, sistema de información geográfica.

ABSTRACT

Scientific community concurs that the use of ontologies is critical for semantic processing, particularly the spatial one. To enhance an ontology, it is needed to incorporate its ontologic components. Making it manually takes the use of an ontology editor, plus having knowledge about ontologic engineering to make the edition. It takes time and effort, and as human beings, there is the chance to make mistakes. In this research is proposed a method for developing a software tool that allows reducing the cost in time and effort of the enrichment of large geographic ontologies, from the analysis of Spatial Databases and semi-automatically. By getting the information from these sources, the ontological components, which subsequently allow the construction of an ontology using the framework Jena are obtained. This ontology is then integrated with the geographic domain ontology *Kaab-Ontology*, incorporating new components to it. A Geographic Information System is an application that makes use of geographic information to provide analysis capabilities, management, query and data visualization. With the development of the method it was possible to minimize the cost in terms of effort and time of the enhancement process of a geographic domain ontology.

Keywords: spatial databases, semantic enrichment, ontology, spatial semantic, geographic information system.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
Introducción.....	5
1.1 Definiciones Ontológicas.....	5
1.1.2 Tipos de Ontologías	7
1.2 Geo-ontología.....	9
1.3 Lenguajes para Representar Ontologías.....	10
1.4 Herramientas.....	12
1.5 Base de Datos Espacial	15
1.7 Estado del Arte.....	16
1.7.1 Las ontologías en la organización y representación de la información.....	18
1.7.2 Ontologías y Base de datos	20
1.7.3 Ontología del Dominio Geográfico	21
1.7.4 Análisis de soluciones existentes	21
Conclusiones Parciales	24
CAPÍTULO 2 SELECCIÓN DE LAS TECNOLOGÍAS. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN	25
Introducción.....	25
2.1 Descripción de la solución que se propone.....	25
2.2 Análisis de la Metodología	26
2.3 Tendencias actuales en el desarrollo de sistemas informáticos.....	28
2.3.1 Lenguajes de Programación.....	31
2.3.2 Marco de Trabajo	33
2.4 Modelo de dominio.....	34
2.4.1 Descripción de los conceptos del modelo de dominio	35
2.5 Especificación de requisitos.....	35
2.5.1 Requisitos funcionales.....	36

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

2.5.2 Requisitos no funcionales	39
2.6 Arquitectura de software	41
2.6.1 Patrones arquitectónicos.....	41
2.7 Modelo de Diseño.....	42
2.7.1 Diagrama de Clase del Diseño	43
2.7.2 Patrones de diseño	46
Conclusiones Parciales	48
CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	49
Introducción.....	49
3.1 Diagrama de componentes	49
3.2 Modelo de Despliegue	50
3.3 Estándares de Codificación	51
3.4 Pruebas del software	52
3.4.1 Tipos de Pruebas	53
3.5 Prueba Realizadas	54
Conclusiones parciales	61
CONCLUSIONES GENERALES	62
REFERENCIAS BIBLIOGRÁFICAS.....	63

ÍNDICE DE TABLAS Y FIGURAS

Tabla # 1 Especificación del RF 21: Construir Ontología.....	37
Tabla # 2 Especificación del RF 22: Enriquecimiento de la ontología del dominio geográfico	38
Tabla # 3 Secciones a probar en el RF16: Construir ontología.....	58
Tabla # 4 Secciones a probar en el RF17: Integrar ontología.	59
Figura 1 Tipos de Ontologías.....	9
Figura 2 Etapas del Ciclo de Vida	28
Figura 3 Modelo Conceptual	35
Figura 4 Diagrama de Clases del diseño del proceso 1. Realizar análisis sobre los conceptos. ...	43
Figura 5 Diagrama de Clases del diseño del proceso 5. Realizar análisis sobre la ontología.....	44
Figura 6 Patrón Creador	47
Figura 7 Diagrama de Componente	50
Figura 8 Modelo de Despliegue	51
Figura 9 Estándares de Codificación.....	52
Figura 10 Método escribirOntologia de la clase ControladoraEventos	55
Figura 11 Grafo que genera el método.....	55
Figura 12 Funcionamiento de la Prueba de Caja Negra	58

INTRODUCCIÓN

Actualmente las Tecnologías de la Información y la Comunicación, también conocida como TIC, tienen un papel fundamental en el desarrollo de la sociedad debido a la acelerada evolución de las mismas. Este progreso ha sido propiciado por la aparición de la tecnología digital, que unida al desarrollo de ordenadores cada vez más potentes, ha permitido a la humanidad progresar rápidamente en la ciencia y la técnica desplegando el arma más poderosa: la información y el conocimiento. Las TIC no solo permite el aumento de los resultados en la ciencia, los servicios y la producción, sino además en nuestra manera de pensar y actuar.

Como parte del propio desarrollo de las TIC surgen los Sistemas de Información Geográfica (SIG), tecnología de manejo de información geográfica formada por equipos electrónicos programados adecuadamente, que permiten manejar una serie de datos espaciales y realizar análisis complejos con estos, siguiendo los criterios impuestos por el equipo científico. Estos sistemas informáticos son capaces de capturar, almacenar, integrar, manipular, analizar y visualizar gran cantidad de datos espaciales (Gottfried, 2014).

En Cuba se desarrollan los SIG con el objetivo de facilitar la gestión y el manejo de la información geográfica en diferentes áreas. La Universidad de Ciencias Informáticas (UCI) cuenta con el desarrollo de la plataforma GeneSIG, proyecto llevado a cabo por especialistas de la Empresa de Tecnologías de la Información para la Defensa (XETID), de la UCI y del grupo empresarial GEOCUBA. La plataforma GeneSIG promueve el uso de componentes reutilizables y tiene implementado un conjunto de funcionalidades que facilitan el desarrollo del SIG, cuyo principal objetivo es favorecer la representación y el análisis de la información geoespacial asociada a negocios específicos.

Debido a la gran cantidad de datos geoespaciales existentes y a la necesidad de que los mismos sean compartidos por el mayor número posible de usuarios, se ha hecho imprescindible el uso de los SIG como una herramienta integradora. Algunos autores (Garea Llano, y otros, 2009) han sugerido que estructuras sofisticadas como las ontologías son buenas candidatas para representar y resumir los datos geográficos, a la par de lograr que los mismos sean compartidos. Las ontologías proveen una comprensión compartida y consensuada del conocimiento de un dominio, que puede ser comunicado entre personas y sistemas heterogéneos. Por ello, el desarrollo de ontologías, se presenta como el instrumento adecuado para

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

alcanzar la integración semántica en el entorno de las infraestructuras de los datos espacial. Es así como se han concebido los SIG de nueva generación, en los que las ontologías sean un componente activo dentro de su arquitectura (Garea Llano, y otros, 2009).

Las ontologías proporcionan la vía para representar el conocimiento, y durante la última década en áreas como la Ingeniería del Conocimiento, Inteligencia Artificial y Ciencias de la Computación, han utilizado las Ontologías en aplicaciones relacionadas con el procesamiento del lenguaje natural, comercio electrónico, integración de información inteligente, consulta de información, integración de bases de datos, bioinformática, educación y en la web semántica. 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 entendimiento común y compartido de algunos dominios que pueden ser comunicados entre las personas y los sistemas aplicados.

Las ontologías pueden ser construidas de forma manual o automática. El hombre en correspondencia de sus habilidades y conocimientos es capaz de construir taxonomías de manera eficiente, pero la construcción de forma manual de una ontología representa un trabajo que implica esfuerzo, debido a que el conocimiento crece cada vez más y es difícil generar, mantener y expandir con adecuada calidad ontologías de gran tamaño. Así pues, se hace necesario automatizar la construcción de ontologías por medio de la aplicación de métodos que permitan minimizar el costo en tiempo y esfuerzo para su desarrollo y mantenimiento.

El **problema a resolver** radica en que: El enriquecimiento de las ontologías geográficas de manera manual es un trabajo agotador en el que hay que emplear mucho tiempo y esfuerzo, debido a que el conocimiento crece y es difícil generar, mantener y expandir con adecuada calidad en ontologías de gran tamaño.

Por tanto en la presente investigación el **objeto de estudio** lo compone el enriquecimiento de ontologías, enmarcado en el **campo de acción** el enriquecimiento de ontologías geográficas.

Se ha definido como **objetivo general**: Desarrollar una herramienta de software que a partir del análisis de bases de datos espaciales y de manera semi-automatizada, permita disminuir el costo en tiempo y esfuerzo del enriquecimiento de ontologías geográficas de gran tamaño.

Objetivos específicos:

1. Realizar el marco teórico y el estado del arte correspondiente al enriquecimiento de ontologías y ontologías geográficas.
2. Proponer un método¹ semiautomático para enriquecer ontologías geográficas a partir del procesamiento de bases de datos espaciales.
3. Implementar un plugin² para GENESIG que desarrolle el método propuesto.
4. Validar la herramienta desarrollada a partir de la realización de pruebas de software.

Preguntas Científicas:

- ✓ ¿Cómo obtener los componentes ontológicos a partir del procesamiento de una base de datos geográfica?
- ✓ ¿Cómo se pueden unir dos ontologías geográficas?
- ✓ ¿Existirán herramientas efectivas para enriquecer una ontología geográfica por medios semiautomáticos?
- ✓ ¿Qué métodos me permitirá procesar los datos de una base de datos espacial?
- ✓ ¿Se podrá elaborar una propuesta teórica eficiente para el enriquecimiento de la ontología geográfica?
- ✓ ¿Cuáles tecnologías y herramientas permitirán enriquecer una ontología geográfica?

En la etapa de investigación del presente trabajo de diploma se usaron métodos teóricos. Estos permiten revelar las relaciones esenciales del objeto de investigación, no observables directamente. Los métodos participan en la etapa de asimilación de hechos, fenómenos y procesos, y en la construcción del modelo (Zayas, 1995).

¹ Método es un modo, manera o forma de realizar algo de forma sistemática, organizada y/o estructurada. Hace referencia a una técnica o conjunto de tareas para desarrollar una tarea.

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 del Procesamiento de Bases de Datos Espaciales.

Análisis y Síntesis: para realizar el estudio del estado del arte enfocado en las ontologías y las bases de datos espaciales, con el objetivo de comprender el funcionamiento de las mismas para el desarrollo de la investigación. Se utiliza para seleccionar los componentes por separado que posteriormente permitirá darle cumplimiento al objetivo general que se propone.

Histórico-lógico: para llevar a cabo el estudio crítico de investigaciones similares y realizar análisis de las mismas para el desarrollo de la investigación.

El presente trabajo consta de introducción, tres capítulos, conclusiones generales y las referencias bibliográficas que complementan el cuerpo del trabajo.

Capítulo 1: Fundamentación Teórica. Se analizan los principales conceptos y componentes relacionados con el tema de investigación, así como las características fundamentales de herramientas y lenguajes existentes para el enriquecimiento de ontologías geográficas de manera semiautomática. Además de un estudio del estado del arte, y de las soluciones existentes.

Capítulo 2: Selección de las Tecnologías. Análisis y Diseño de la Solución. Se realiza el análisis y diseño de la solución propuesta. Se describe el modelo de dominio, permitiendo una mejor comprensión de la solución. Se enumeran y detallan los requisitos funcionales y no funcionales. También se generan los diagramas que permitirán un mayor entendimiento de las funcionalidades de la herramienta.

Capítulo 3: Implementación y Validación de la Solución Propuesta. Se presentan los elementos fundamentales del proceso para enriquecer las ontologías de manera semiautomática, donde se hace un análisis de los artefactos como el modelo de despliegue y el diagrama de componentes. Además, se realizan pruebas funcionales de caja blanca, caja negra; y el respectivo análisis de las mismas. Finalmente, la prueba de aceptación para validar el resultado de la solución propuesta.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se abordan varios conceptos relacionados con las ontologías, los cuales son de suma importancia para la presente investigación. Además se muestran los distintos tipos de ontologías que existen, así como las características de algunos lenguajes y herramientas utilizadas para la construcción de las mismas; complementando todo esto, con un estudio del estado del arte y de las soluciones existentes para el desarrollo de la solución se propone.

1.1 Definiciones Ontológicas

Las ontologías han sido definidas de diferentes formas, pero generalmente sus autores defienden las ontologías como un medio de entendimiento entre las personas y los sistemas informáticos. Es común que cada comunidad que desarrolla ontologías adopte una definición propia dependiendo de sus necesidades. Entre las tantas definiciones que se pueden encontrar, la más aceptada es la propuesta por (Gruber, 1995): “Una Ontología es una especificación formal y explícita de una conceptualización compartida. Tienen un papel clave para estructurar, manejar, compartir y procesar la información en la Web Semántica, corpus digitales, comercio electrónico y aplicaciones médicas; ya que proveen conocimiento compartido y común de un determinado dominio”.

Las ontologías incluyen una completa generalización/especificación de sus clases y subclases, las cuales están formalmente especificadas (incluyendo sus relaciones e instancias) asegurando la consistencia en los procesos deductivos y estas son implementadas en un lenguaje específico de representación Ontológica de manera que la especificación de sus clases, relaciones entre estas y sus restricciones dependerán de las características de dicho lenguaje.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Los términos utilizados en esta definición se basan principalmente en lo siguiente:

- ✓ **Conceptualización:** Modelo abstracto de un fenómeno, que puede ser visto como un conjunto de reglas informales que restringen su estructura. Por lo general se expresa como un conjunto de conceptos (entidades, atributos, procesos), sus definiciones e interrelaciones.
- ✓ **Formal:** Organización teórica de términos y relaciones usados como herramienta para el análisis de los conceptos de un dominio.
- ✓ **Compartida:** Se refiere a la captura del conocimiento consensual que es aceptado por una comunidad.
- ✓ **Explícita:** Conciernen a la especificación de los conceptos y a las restricciones sobre estos.

Posteriormente han existido otras definiciones, (Hendler, 2001) plantea que: “Una Ontología es un conjunto de términos de conocimiento, que incluye un vocabulario, relaciones y un conjunto de reglas lógicas y de inferencia sobre un dominio en particular”. La importancia de esta definición son las relaciones y el conjunto de reglas, expresando que las Ontologías describen el significado de las relaciones entre conceptos y permiten de alguna manera formas de razonamiento.

Otros autores como (Ehrig, 2007) utilizó la definición que fue desarrollada por el grupo de manejo de conocimientos del Instituto AIFB de la Universidad de Karlsruhe. La cual plantea que una ontología contiene un núcleo donde es referenciada como un esquema, además de estar compuesta por axiomas, una base de conocimiento y un lexicón³, no de la misma forma planteada por (Euzenat, y otros, 2007), donde lo define como tipos de entidades que están presentes en los lenguajes de ontologías, con los cuales se expresan. La más práctica es la propuesta por (Hess, 2008), definiéndola como una 4-tupla $O = \langle C, P, I, A \rangle$ 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.

³ Conjunto de las palabras y lexemas de una lengua y libro en que se contienen: lexicón informatizado.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Un concepto $c \in C$ es cualquier fenómeno de interés a ser representado en la ontología y es definido por un término t que es usado como su nombre. El nombre de un concepto está dado por una función unaria $t(c)$.

Una propiedad $p \in P$ es una componente que es asociada 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 $i \in I$ es una ocurrencia de un concepto c , con un valor para cada propiedad p asociada al concepto y un único identificador.

Un axioma 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.

Haciendo un análisis de la última definición planteada, se puede señalar el tratamiento que le da a las definiciones. Se refleja el componente relación de las dos primeras definiciones como el componente propiedad de la última definición de tipo objeto. En las dos primeras definiciones, las relaciones son tomadas como componente fundamental de las ontologías y no como un caso específico de la componente propiedad. También en la última definición se refleja la generalización de la relación como axioma a la relación jerárquica planteada en las dos primeras definiciones, considerándose como otra especialización de la componente axioma, lo cual no es así ya que la relación de taxonomía es una componente necesario en las ontologías.

1.1.2 Tipos de Ontologías

Las ontologías pueden clasificarse de acuerdo a la cantidad y tipo de estructura de la conceptualización en (Fensel, 2001):

- ✓ Ontologías terminológicas, que especifican los términos que son usados para representar conocimiento en el universo del discurso. Suelen ser usadas para unificar vocabulario en un campo determinado.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

- ✓ Ontologías de información, especifican la estructura de almacenamiento de bases de datos. Ofrecen un marco para almacenamiento estandarizado de información.
- ✓ Ontologías del modelado del conocimiento, especifican conceptualizaciones del conocimiento. Contiene una rica estructura interna y suelen estar ajustadas al uso particular del conocimiento que describen.

Según su dependencia y relación con una tarea específica desde un punto de vista, (Guarino, 1998) clasifica las ontologías en: (Ver figura 1)

- ✓ Ontologías de Alto nivel o Genéricas: describen conceptos más generales, por ejemplo, el espacio, tiempo, materia, objeto, evento, acción, los cuales son independientes de un problema o dominio en particular.
- ✓ Ontologías de Dominio: describen un vocabulario relacionado con un dominio genérico, por ejemplo, medicina, o una tarea o actividad genérica, como diagnóstico, especializando los términos introducidos en la ontología de alto nivel.
- ✓ Ontologías de Tareas o Técnicas básicas: describen una tarea, actividad o artefacto, por ejemplo componentes, procesos o funciones.
- ✓ Ontologías de Aplicación: describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.

La clasificación sobre la base de las ontologías se concibe como una categorización de la información. En esta, la información fluye por medio de categorías, como expresión de clases que agrupan conceptos a partir de su similitud: palabras en general que componen los contenidos discursivos y palabras específicas que determinan en esos contenidos los conceptos particulares. Se utilizará una ontología de dominio donde se describirán los términos del dominio definiendo sus conceptos y relaciones, para de esta forma lograr una interoperabilidad semántica de la información que se maneja. Las ontologías de dominio establecen la base conceptual que hace posible la anotación semántica y constituyen un acuerdo común sobre un dominio. Se escoge este tipo de ontología para enriquecer la misma, ya que este fenómeno posibilita reflejar con mayor profundidad las propiedades esenciales de los conceptos y que a la vez estos se enriquezcan. Además, cuando una entidad se asigna a una categoría se crea un contexto (dominio) que no solo ofrece información sobre la entidad, sino también sobre las formas de interactuar con ella.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

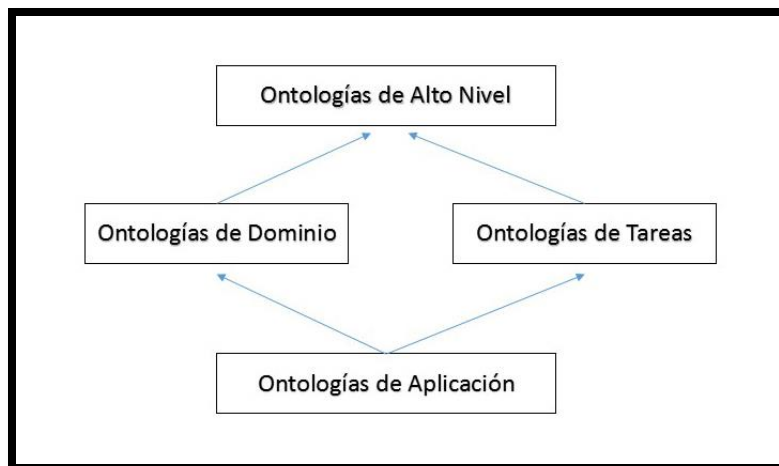


Figura 1 Tipos de Ontologías.

1.2 Geo-ontología

La actividad cotidiana de la sociedad implica el surgimiento de grandes volúmenes de información entre los cuales se encuentran los nuevos conceptos, lugares, ideas, términos, nombres geográficos, en fin, datos que en pocas palabras se traducen como reproducción de conocimiento humano. Una persona que tenga la necesidad de recuperar o acceder a diversos recursos puede encontrar un mayor grado de dificultad o ambigüedad al momento de ubicar la información, debido al gran volumen de información y a los complejos medios con los cuales se recupera la información en la actualidad.

De aquí viene dada la necesidad de generar un sistema que permita recuperar datos geográficos contruidos por medio de descriptores y conceptos que van a servir para desarrollar el proyecto de una ontología geográfica. Las ontologías que tratan la temática geoespacial, contienen características particulares de este campo de estudio. Debido a su nivel de especialización, a estas ontologías geográficas se les han denominado geo-ontologías.

Estas cumplen con todas la características de una ontología convencional, pero tienen además propiedades propias de este dominio, por ejemplo, un par de coordenadas (x,y) que representa la posición geográfica de un objeto (Vera, y otros, 2009). Las geontologías son estructuras suficientemente adaptables donde la integración de distintos tipos de información es posible. Su uso proporciona un salto cualitativo en funcionalidad y posibilidad, permitiendo la comunicación entre personas, organizaciones y aplicaciones. Son

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

estructuras que pueden crecer, integrarse y reutilizarse en otros dominios, estas características constituyen beneficios de las geontologías como subconjunto de las ontologías. Una geontología se define como una ontología que se delimita a relacionar información, datos y metadatos geográficos, con el objetivo fundamental de lograr la interoperabilidad semántica entre ellos.

Los autores (Oliva Santos, y otros, 2007) determinan que una geontología es una ontología que ofrece una descripción de entidades geográficas y difiere de otras ontologías por la presencia predominante de relaciones topológicas que caracteriza a la información espacial. Una geontología tiene dos tipos básicos de conceptos, los conceptos físicos y los conceptos sociales.

Según (Oliva Santos, y otros, 2007) las geontologías son estructuras en las que es posible la integración o asociación dato-conocimiento. Se explica también que las anotaciones de los objetos geográficos se modelarán como instancias de conceptos pertenecientes a geo-ontologías, de modo que hay que distinguir de cual ontología se está haciendo referencia.

1.3 Lenguajes para Representar Ontologías

Las Ontologías para ser expresadas requieren de un lenguaje lógico y formal. En la inteligencia artificial se han desarrollado numerosos lenguajes para este fin, algunos basados en la lógica de predicados, como KIF y Cycl que ofrecen poderosas primitivas de modelado, y otros basados en *frames* (taxonomías de clases y atributos), que tienen un mayor poder expresivo, pero menor poder de inferencia; e incluso existen lenguajes orientados al razonamiento. Todos estos lenguajes han servido para desarrollar otros lenguajes aplicables a la Web. En un lenguaje de Ontologías se pretenderá un alto grado de expresividad y uso.

Plantean (Ramos, y otros, 2007) que los lenguajes para codificar Ontologías deben contemplar ciertos aspectos tales como son: sintaxis bien definida, semántica específica, suficiente expresividad, fácilmente traducible entre los lenguajes ontológicos y permitir 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 (W3C⁴) recomienda RDF y OWL.

⁴ Es una comunidad internacional donde los estados miembros trabajan para poder desarrollar estándares para el desarrollo web y así ayudar a un mejor desarrollo del Internet a nivel mundial.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Entre los principales lenguajes de ontologías se destacan los siguientes:

SHOE: *Simple HTML Ontology Extensions*. Fue el primer lenguaje de etiquetado para diseñar ontologías en la Web. Este lenguaje nació antes de que se ideara la Web Semántica. Las ontologías y las etiquetas se incrustaban en archivos HTML. Este lenguaje permite definir clases y reglas de inferencia, pero no negaciones o disyunciones.

OIL: *Ontology Inference Layer*. Este lenguaje, derivado en parte de SHOE, fue impulsado también por el proyecto de la Unión Europea On-To-Knowledge. Utiliza ya la sintaxis del lenguaje XML y está definido como una extensión de RDFS. Se basa tanto en la lógica descriptiva (declaración de axiomas) y en los sistemas basados en *frames* (taxonomías de clases y atributos). OIL posee varias capas de sub-lenguajes, entre ellas destaca la capa base que es RDFS, a la que cada una de las capas subsiguientes añade alguna funcionalidad y mayor complejidad. La principal carencia de este lenguaje es la falta de expresividad para declarar axiomas.

OWL: (*Web Ontology Language*) o Lenguaje de Ontologías para la Web es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la Web. Se trata de una recomendación del W3C, y puede usarse para representar ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre aquellos términos (ontologías). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste. Se trata de un lenguaje diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones.

La característica que define a cada lenguaje es su expresividad, se explica a continuación:

- ✓ **OWL Lite:** Es el sub-lenguaje con sintaxis más simple, su intención es ser utilizado en situaciones donde se requiera una jerarquía de clases y restricciones simples.
- ✓ **OWL DL:** Es mucho más expresivo que OWL Lite y está basado en lógica descriptiva. Proporciona la máxima expresividad posible sin perder la completitud computacional (todas las conclusiones pueden ser deducidas) y la posibilidad (todos los cálculos se realizan en un tiempo finito).

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

- ✓ **OWL Full:** Es el sub-lenguaje más expresivo, su intención es ser utilizado en situaciones donde una alta expresividad es más importante que la capacidad de garantizar la completitud computacional y la posibilidad.

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 (Fensel, 2001). El modelo de datos de RDF está formado por recursos (objetos) y pares de atributos/valores. Un recurso representa cualquier entidad que pueda ser referenciada por un URI (Identificador Único de Recursos). Los atributos representan las propiedades de los recursos, y sus valores pueden ser entidades atómicas (por ejemplo: *strings*, *int*) u otros recursos. Un modelo RDF puede ser representado como un grafo dirigido, donde los recursos y los valores constituyen los nodos, y los atributos constituyen los arcos formando una red semántica.

TURTLE: (Terse RDF Triple Language) es un formato para la expresión de los datos en el *Resource Description Framework* (RDF). Fue la primera notación aceptada por la W3C, en Agosto de 2011. Hoy en día es probablemente la notación RDF más extendida en la web semántica. Utiliza definición de prefijos para aumentar la legibilidad.

1.4 Herramientas

En correspondencia al estudio realizado sobre el estado del arte se han podido identificar una serie de herramientas fundamentales para el trabajo con ontologías. Estas herramientas se organizan en las categorías siguientes: Desarrollo, Evaluación, Combinación e integración, Herramientas de anotación, Almacenamiento y consulta, y Aprendizaje. La principal herramienta para la construcción de una ontología son los editores. Los editores de ontologías soportan: la definición de jerarquías de conceptos, la definición de atributos o propiedades de los conceptos, y la definición de axiomas y restricciones. Estos suelen ser desarrollados para un tipo de lenguaje específico, pero muchos de ellos incorporan módulos para soportar otros lenguajes de especificación diferentes.

A continuación se destacan algunas:

JENA: Es un framework Java para el desarrollo de aplicaciones de la Web Semántica. Ofrece un entorno para RDF, esquemas RDF y OWL e incluye un motor basado en reglas de inferencia. Jena incluye un motor

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

de inferencia basado en reglas para llevar a cabo un razonamiento basado en ontologías OWL y RDFS, y una variedad de estrategias de almacenamiento para almacenar RDF. Es similar a la de sésamo; sin embargo, a diferencia de sésamo, Jena proporciona soporte para OWL (Lenguaje de Ontologías Web). El marco tiene varios razonador interna y el razonador Pellet (una fuente abierta Java OWL-DL razonador) se puede configurar para trabajar en Jena. Es además una API Java que se puede utilizar para crear y manipular gráficos RDF y también tiene clases de objetos para representar gráficos, recursos, propiedades y literales. Las interfaces que representan recursos, propiedades y literales se llaman recursos, la propiedad y Literal respectivamente. En Jena, un gráfico que se llama un modelo y está representado por el modelo de interfaz.

WebOnto: Es una herramienta que proporciona la visualización, navegación y edición de soporte basado en la web para el desarrollo y el mantenimiento de ontologías y modelos de conocimiento especificadas en OCML. Una ontología puede ser vista como un modelo de la estructura conceptual de un dominio y WebOnto proporciona los medios para representar gráficamente esta. WebOnto es un applet de Java junto con un servidor web personalizado que permite a los usuarios navegar y editar modelos de conocimiento a través de Internet. WebOnto ya está disponible como un servicio público. WebOnto se ha construido como parte de los proyectos Patman, HCREMA y Enrich. Además de estos proyectos WebOnto se está utilizando actualmente en el PlanetOnto, ScholOnto.

OntoEdit: Es una herramienta de edición de ontologías que apoya el desarrollo y mantenimiento de las mismas utilizando medios gráficos en un entorno web. Permite la representación semántica de lenguajes conceptuales y estructuras mediante conceptos, jerarquías de conceptos, relaciones y axiomas. El interfaz es abierto y permite a los usuarios ajustar OntoEdit a sus necesidades. Además tiene disponibles una gran variedad de funcionalidades fáciles de usar y adaptables a los diferentes escenarios. Me permite representar gráficamente ontologías y además, almacenarlas y posteriormente manipularlas en una base de datos relacional. Se le pueden añadir nuevas funcionalidades también a través de *plugins* que le aportan interoperabilidad permitiendo interactuar con otras herramientas como Sesame.

Protégé: Fue desarrollado por *Stanford Medical Informatic* (SMI) en la Universidad de Stanford siendo este uno de los editores más utilizados y más conocido. Es un editor de ontologías de código abierto para construir ontologías sobre RDFS, OWL y XML Schema. Herramienta basada en el lenguaje Java. Sus capacidades gráficas facilitan la edición de ontologías. Actualmente es uno de los editores de ontologías

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

más usado por investigadores para desarrollar sus ontologías, ya que es una herramienta que se actualiza con bastante regularidad y a la que se le pueden añadir módulos y *plugins* con nuevas funcionalidades, entre los que destaca OWLViz, herramienta de visualización gráfica de ontologías basada en grafos. Permite que la ontología desarrollada se exporte a los diferentes lenguajes de especificación más empleados actualmente (RDF, OWL, etc.).

OntoLingua: Desarrollado en 1990 por el Laboratorio de Sistemas de Conocimiento, KSL (*Knowledge Systems Laboratory*) de la Universidad de Stanford, está orientado al lenguaje Ontolingua, aunque posteriormente incluyeron otros lenguajes. Proporciona un entorno distribuido y colaborativo para la creación, edición, modificación, navegación y utilización de ontologías mediante la Web. También incluye una interfaz de programación de aplicaciones (API) para poder integrar las ontologías del servidor con agentes preparados para Internet.

Después de haberse evaluado las herramientas anteriores, se llegó a la conclusión de que la más recomendable a usar es Jena, basado en sus características y ventajas como se explica a continuación:

El marco Jena crea una capa adicional de abstracción que traduce las declaraciones y construcciones de la Web Semántica en artefactos Java, como clases, objetos, métodos y atributos. Estos artefactos reducen el esfuerzo necesario para la programación de aplicaciones de la Web Semántica. Uno de los lados más fuertes de Jena reside en su excelente documentación. Los recursos exhaustivos, incluyendo descripciones y tutoriales que se pueden encontrar en la web, animan a los programadores a desarrollar sus aplicaciones de Web Semántica utilizando este marco.

Jena también apoya el trabajo con el RDF Schema, proporcionando API⁵ para todas las extensiones de vocabulario que trae. Por otra parte, cubre el uso de OWL en una de las tres variantes: Completo, Descripción *Logic* y *Lite*. La API de OWL ofrece la posibilidad de navegar a través de la gráfica, localizar recursos y recuperarlos a partir del modelo. La API que viene con el marco, hace que el proceso de intercambio de conocimientos sea más fácil, ya que todos los recursos vienen con su URI.

⁵ Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés *Application Programming Interface*. En español sería *Interfaz de Programación de Aplicaciones*.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Debido a su alto nivel de los genéricos elaborados en su diseño de software, Jena puede ser fácilmente vinculado a bases de datos SQL de diferentes proveedores. Todos los que un desarrollador necesita es un controlador apropiado para la base de datos SQL en particular. Por otra parte un componente poderoso del Marco Jena es la API de inferencia.

1.5 Base de Datos Espacial

Las Bases de Datos Espaciales (BDE) constituyen una fuente de información geográfica muy útiles para la generación, identificación de instancias, de conceptos y relaciones espaciales. A continuación algunos aspectos importantes sobre la misma.

Conceptos sobre BDE:

- ✓ DBMS⁶ con capacidades para representar y manipular datos con una referencia en el espacio⁷.
- ✓ Esta base de datos comprende la asociación ente sus dos principales componentes: datos espaciales⁸ y atributos o datos no espaciales⁹.
- ✓ El lenguaje de consulta (SQL) es extendido para manejar la geometría de los datos a través de puntos, líneas y polígonos y son incorporadas nuevas funciones que permiten la selección y recuperación de los datos tanto por criterios alfanuméricos como geométricos.
- ✓ Un concepto de NCGIA¹⁰, sugiere que una BDE "es una colección de datos referenciados en el espacio que actúa como un modelo de la realidad".

Integrar las distintas BDs heterogéneas con los distintos nombres de tabla, de campo, e incluso distinto número de campos, es una motivación, además de buscar la interoperabilidad entre las distintas Bases de Datos. Investigaciones de (Sarabia, 2008) concluyen que el espacio establece un marco de referencia para definir la localización y relación entre objetos. El que normalmente se utiliza es el espacio físico, el cual es un dominio manipulable, perceptible y que además sirve de referencia. La construcción de una base de

⁶ *Data Base Management System son las siglas en inglés para los Sistemas de Gestión de Bases de Datos (SGBD).*

⁷ *Establece un marco de referencia para definir la localización y relación entre objetos.*

⁸ *Dato que se ubica en el espacio, ya sea con referencia a un sistema de coordenadas o a un orden topológico (ubicación, dimensión, forma).*

⁹ *Son las características cuantitativas asociadas al objeto que se desea describir (Tablas con atributos E-R).*

¹⁰ *National Center for Geographic Information and Analysis.*

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada, que pueda ser procesada por el lenguaje utilizado por las computadoras. Se puede entender como BDE a la colección de los datos de los objetos geográficamente referenciados que representan la realidad de una zona geográfica.

Con el análisis de la BDE se concluye: que no solo sirven para modelar mapas, se utilizan en una amplia gamas de aplicaciones explotando los datos mediante sistemas de información geográficas. Estos sistemas de información geográfica apoyan a mostrar la información un tanto gráfico de la extracción de datos espaciales, y son utilizados además para la generación de capas en los mapas, un ejemplo es Google Maps.

1.7 Estado del Arte

Este tópico se define con el objetivo de adquirir los conocimientos más actualizados en relación con el enriquecimiento de las ontologías y en particular las ontologías geográficas.

Las ontologías hoy en día son herramientas computacionales que permiten la administración de la semántica de los datos, generando nuevos conocimientos a partir del uso de razonadores. Múltiples investigadores han centrado sus estudios e investigaciones en las ontologías, por la gran importancia que estas representan. Entre ellos se encuentran: (Torres, 2007), (Torres Rodríguez, 2008), (Peinado Gil, 2008), (García Sánchez, 2009) y (Macías, 2003).

Los SIG no están exentos a esta nueva concepción de manejo de información y debido a la gran cantidad de datos almacenados referentes a la Tierra, ha surgido un contexto propicio donde la comunidad científica internacional ha manifestado su preocupación por el desarrollo de tecnologías para su unificación, y de las herramientas para el manejo y análisis de dichos datos.

El desarrollo de ontologías, se presenta como el instrumento adecuado para alcanzar la integración semántica en el entorno de las infraestructuras de los datos espaciales de una manera mucho más abstracta, en la que el conocimiento juegue un papel fundamental. Es así como se han concebido los SIG de nueva generación, en los que las ontologías son un componente activo dentro de su arquitectura. Estos Sistemas de Información Geográfica Gobernados por Ontologías (SIGGO) son construidos utilizando clases derivadas de las mismas, y como resultado es posible utilizar todo el conocimiento embebido; estas clases

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

contienen atributos y funciones que darán soporte a todas las funcionalidades del sistema. La nueva generación de SIGGO deberá ser capaz de resolver la interoperabilidad semántica, en la cual un hecho puede tener un valor mayor que solamente su descripción (Oliva Santos, y otros, 2009).

La concepción de los SIGGO como fase superior a los SIG, interviene en este proceso como un medio integrador que obvia la clasificación de los datos a través de su forma de representación. Estos sistemas utilizan el punto de vista de la expresividad semántica para de una forma natural integrar los distintos tipos de información a través de entes flexibles, que comúnmente, se encuentran basados en estándares y por consecuente son cimentados en ontologías.

El campo de investigación en torno a las ontologías en estos momentos es uno de los más aplicables en las tecnologías de varios países. En Cuba por ejemplo, se destaca la labor del Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), surgido en el 2004 y está orientado a investigaciones teóricas y aplicadas al área de Reconocimiento de Patrones (RP) y Minería de Datos (MD).

Numerosos profesionales cubanos en estos momentos son de los más destacados con sus pesquisas en el área de las ontologías, geo-ontologías y semántica espacial, entre ellos se encuentran: Dr. Eduardo Garea Llano, jefe del Departamento de RP del CENATAV, Dr. Rainer Larín Fonseca y Lic. Francisco Vera Voronisky, ambos investigadores del Departamento de RP del CENATAV, y Dr. Rafael Oliva Santos, Profesor Principal Asistente (PPA) del Departamento de Computación, Facultad de Matemática y Computación (MATCOM), Universidad de La Habana (UH).

Además, Cuba como país en vías de desarrollo no se encuentra ajena a la aplicación de las TIC. Para ello, desde hace algunos años se lleva a cabo el Programa Nacional para la Informatización de la Sociedad Cubana (PNISC), cuyo objetivo principal es acercar a la sociedad cubana al manejo adecuado de las TIC. El bloqueo económico impuesto impide que se tenga mayor desarrollo en este campo, lo cual afecta al desarrollo de la economía cubana. A pesar de esto la UCI, nacida al calor de la Batalla de Ideas, visionada por el Comandante en Jefe Fidel Castro como “universidad de excelencia”, forma parte activa del PNISC y tiene, la misión de producir software de calidad, tanto para clientes nacionales como internacionales. En la misma se encuentra el Centro de Desarrollo Geoinformática y Señales Digitales (GEYSED), el mismo cuenta en su estructura con varias líneas de investigación y desarrollo, entre las cuales se encuentra la

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

línea de Manejo de Información Geográfica, la cual se especializa en la implementación de la plataforma soberana GeneSIG, que es un producto utilizado para el desarrollo de SIG.

La necesidad de compartir e intercambiar información se han logrado en buena parte gracias a los SIG, que son capaces de manejar datos geográficos almacenados. Pero sucede que la interoperabilidad semántica de la información en los SIG aún no ha sido alcanzada. Por la gran complejidad y abundancia de información contenida en los datos geográficos brota una serie de problemas que específicamente se enfocan en la interoperabilidad dentro de ellos. Las estructuras que permiten manejar esos eventos son las geontologías, que como subconjunto de las ontologías están preparadas en el contexto de los datos geográficos.

1.7.1 Las ontologías en la organización y representación de la información

Las ontologías en el sentido, de que mientras más ordenada y mejor procesada se encuentre la información, será más fácil desarrollar la selección y posterior adquisición. Por tanto, es interesante la definición de una ontología (González, 2006) como “un instrumento de organización y representación del conocimiento que permite hacer explícitas las reglas implícitas de una parte de la realidad. Idealmente, su presentación formalizada permite que estas declaraciones explícitas sean independientes del sistema que las utiliza y que, a su vez, pueda reutilizarse por otros sistemas”.

Las ontologías significan una evolución de los instrumentos de representación y organización del conocimiento. En estos momentos, sólo se vislumbra este acontecimiento en el medio digital, como expresión de la incesante búsqueda de nuevos instrumentos para mejorar la organización y recuperación de la información en la red y en los servicios tradicionales. Una ventaja de los lenguajes ontológicos, es el hecho de que ellos no desarrollan sólo temas o vocabularios que asocien recursos.

Estos instrumentos como repertorios del conocimiento compartido, retoman aspectos teóricos que tienen sus bases en las raíces de la profesión de información:

- ✓ Son sistemas que establecen clases generales como sinónimo de representación más abstracta del dominio y subclases más específicas que pueden utilizarse en función del grado de profundidad de la descripción.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

- ✓ Definen los términos básicos y en función de ellos, se esbozan las relaciones que se pueden establecer en ese dominio y las reglas para su combinación.
- ✓ Contemplan las restricciones (slots) para delimitar las cualidades de los conceptos y las facetas como expresión de los valores que puede adoptar un slot.
- ✓ Establecen las instancias como objetos de la clase.

También, implican los procesos de análisis y síntesis. El primero, para descomponer el concepto en sus elementos y el segundo, para componerlos. Así, en función de las relaciones descritas, se perfilan las relaciones inversas, transitivas y simétricas para representar estos conceptos en toda su magnitud. Estos procesos adquieren una gran trascendencia para la conformación de las inferencias, porque garantizan que estas no sean erróneas. Por su mecanismo de funcionamiento, estas estructuras cognitivas, abren un espacio para la elaboración futura de resúmenes automatizados, en tanto no transitan sólo de lo general a lo particular, sino que permiten también la relación inversa y esto puede llevar a nuevas aplicaciones futuras de las ontologías en el entorno Web.

Las ontologías se convierten así en una herramienta que permite almacenar el conocimiento para luego garantizar su recuperación en sistemas automáticos capacitados para efectuar disímiles deducciones a partir de esa variedad de relaciones entre los conceptos. Además, producto de la indización puede obtenerse un mayor número de puntos de acceso a la información y concebir estratégicamente servicios con un mayor valor agregado. El almacenamiento, como resultado del procesamiento de la información, integra elementos nunca antes concebidos. En este ámbito, actualmente, no tiene sentido almacenar varias copias de determinado documento y con ello, se rompe otro esquema del que ha partido la selección y adquisición y que ha incidido en este proceso. Además, el almacenamiento constituye la base inmediata para la recuperación, transferencia, reutilización y diseminación del conocimiento, con varias funciones simultáneamente.

Las ontologías por tanto, fungen como herramientas tecnológicas que:

- ✓ Facilitan e intervienen en la selección y adquisición de la información.
- ✓ Enriquecen el procesamiento de la información, a partir de una descripción semántica más profunda.
- ✓ Propician un almacenamiento que sienta nuevos espacios para la búsqueda, recuperación y diseminación efectiva de la información.

1.7.2 Ontologías y Base de datos

Las ontologías están compitiendo en el crecimiento y la popularidad entre los esquemas de bases de datos, debido a las nuevas tecnologías relacionadas con las ontologías que se están desarrollando, especialmente en relación con la Web Semántica. En este punto, los desarrolladores y los usuarios han comenzado a preguntarse acerca de que tecnología se ajusta más a sus necesidades, o incluso si vale la pena cambiar su sistema actual, basado principalmente en las bases de datos relacionales, a favor de la nueva. En la actualidad existe una gran controversia en cuanto a cuál es la mejor tecnología para la representación de la información y el conocimiento en computadoras. En consecuencia, ambas tecnologías han sido comparados con el fin de establecer su grado de similitud, junto con los pros y los contras. Esta discusión, sin embargo, toma varios elementos en cuenta, tales como problemas de eficiencia, capacidad de representación, las tecnologías involucradas, repercusiones en los sistemas reales, etc.

Web Semántica y bases de datos: Varias tecnologías se han desarrollado (ASP, JSP, PHP) para representar páginas web dinámicas y generar sus contenidos utilizando los datos almacenados en las bases de datos en línea. Por un lado, el contenido de estas bases de datos sólo se muestra cuando se realiza una consulta en la base de datos, y por el otro, la descripción semántica de la base de datos se representa mediante su esquema, a menudo no está disponible o incluso inútil porque no puede ser explotada dependiendo del formato elegido para representarla. Las tendencias actuales intentan relacionar bases de datos y Web Semántica para recuperar resultados más enriquecidos, porque la mayoría de los datos se representan en las bases de datos. La comunicación entre ambas tecnologías implica el uso de ontologías que establecen las relaciones apropiadas.

Las Ontologías y las bases de datos pueden representar la misma realidad, pero dependiendo del problema a ser resuelto o la aplicación a ser desarrollado. Las bases de datos relacionales tienen gran hegemonía en el manejo de la información, y se han desarrollado para gestionar otros tipos de datos pudiéndose hacer deducciones, extraer conocimiento o es propietario de objetos multimedia, etc. Estas tecnologías requieren procedimientos de especialización e integración. Las bases de datos son altamente especializados para el manejo eficiente de un determinado tipo de datos. Sin embargo, las ontologías proporcionan un marco de restricción de la libertad de representar una realidad legible por máquina, incluso en la Web. Este marco asume un mundo abierto en el que la información se puede definir de forma explícita, compartida, reutilizada

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

o distribuida. Por otra parte, la información también se puede intercambiar y utilizar para hacer deducciones o consultas.

Para representar la información y ser compartido en la Web, una ontología, proporciona una buena solución. Sin embargo, probablemente implicaría el uso de ambas tecnologías. Una gran cantidad de datos necesita ser almacenada y gestionada correctamente. Las ontologías son una excelente manera de representar la realidad y la base de datos es sin duda el mejor método para almacenar dicha información cuando se trata de un tamaño considerable (Martinez Cruz, y otros, 2011).

1.7.3 Ontología del Dominio Geográfico

El proceso de enriquecimiento se realizará a partir del procesamiento de fuentes de información que serán posteriormente analizadas. Varios investigadores han abordado la construcción de una ontología del dominio geográfico que sirva de guía para la construcción de ontologías de menor nivel (Ping, y otros, 2009), (Tomai, y otros, 2005), (Torres, 2007), (Lopez, y otros, 2010), (Smart, y otros, 2007). Por tal motivo, en esta propuesta no se pretende construir un nuevo modelo ontológico para el dominio geoespacial, sino elegir uno de los que fueron desarrollados y tomarlo como base para la integración con la ontología que se crea a partir del procesamiento de la base de datos espacial. Luego de analizar varias propuestas se seleccionó la ontología del dominio geográfico *Kaab-Ontology* especificada en (Torres, 2007). La misma conceptualiza el dominio geográfico a través del método GEONTO-MET.

1.7.4 Análisis de soluciones existentes

Las ontologías actualmente son objeto de estudio de diferentes investigadores alrededor del mundo. La base de la presente investigación parte de la revisión exhaustiva tanto de los trabajos realizados (artículos, libros, tesis) como las soluciones planteadas por estos autores. A continuación se analizan soluciones existentes que giran alrededor del problema que se plantea en la presente investigación.

Método de Enriquecido Semántico para la Integración de Objetos Geoespaciales (Larin Fonseca, y otros, 2013)

En este trabajo se propone un nuevo método de enriquecido semántico para mejorar el proceso de integración de objetos geoespaciales a través del uso de un nuevo tipo de ontología. Lo resultados obtenidos

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

resultan prometedores para la integración de fuentes heterogéneas de objetos geoespaciales y la realización de análisis sobre estos al nivel semántico.

El método que proponen no se relaciona con el propuesto en la investigación, pero será objeto de análisis el resultado de este trabajo, ya que lo aplican a un SIG para comprobar la validez.

Procedimiento para la obtención de un modelo ontológico para representar la información contenida en bases de datos (Vega Ramírez, y otros)

En este trabajo se propone un procedimiento a través del cual un razonador evalúa la información de una base de datos y la clasifica de forma automática en conceptos, relaciones, roles y atributos. Esta clasificación se desarrolla mediante un procedimiento dividido en dos métodos: Un algoritmo de migración el cual genera una ontología con los elementos del esquema relacional de la base de datos, y una estrategia de clasificación de la información, basada en una serie de consultas *SPARQL* mediante las que se clasifica la información de la base de datos.

Los autores de la presente investigación no concuerdan con la idea plasmada en el resumen del artículo, donde se dice que los razonadores evalúan la información de una base de datos y la clasifican de forma automática en componentes ontológicos. Se considera acorde a la literatura estudiada, que los razonadores se aplican a las ontologías, no a las bases de datos.

Meta-ontología Geoespacial: Ontología para Representar la Semántica del Dominio Geoespacial (Tolaba, y otros, 2013)

En este trabajo se presenta una meta-ontología geoespacial que permite conceptualizar las particularidades de los objetos geoespaciales, a partir de la cual se pueda crear posteriormente una geontología de un dominio dado usando un Base de Datos Espacial.

La meta-ontología que se desarrolla es importante debido a que puede ser usada para dar semántica a los objetos geoespaciales, considerando las características particulares de estos; pero en su desarrollo detectaron la necesidad de un proceso para la instanciación debido a que las relaciones topológicas se establecen entre instancias de los conceptos identificados. También la limitación de no poder modelar la temporalidad que los objetos espaciales pueden tener.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Para el método que se propone en la presente investigación, el tratamiento de las instancias es fundamental para construir la ontología a partir de la base de datos espacial. En este trabajo no se extraen las instancias, la detectan como una necesidad, por ende no utilizaremos este tipo de ontología. Sin embargo el análisis de algunos aspectos de interés, como las herramientas que se utilizaron, impulsó el uso de Protégé para representar la ontología que se construye en este trabajo.

Building Geospatial Ontologies from Geographical Databases (Baglioni, y otros)

En este trabajo se describe un método para la extracción de una ontología geoespacial de los datos geográficos almacenados en base de datos espaciales. Para proporcionar la semántica geoespacial introducen nuevas relaciones que definen la ontología geoespacial que puede servir como base para un sistema de consulta de usuario avanzado. Presentan algunos ejemplos de uso de la metodología en el ámbito urbano. También demuestran que, mediante la adición de una capa de abstracción semántica, pueden hacer referencia a conceptos que no tienen una correspondencia directa con la tabla de base de datos. Esto le da al usuario una mayor fuerza expresiva y una vista semántica a los datos geográficos.

Mediante el estudio de este trabajo, comprendimos la importancia que tienen el uso de las ontologías, el significado y el funcionamiento del mismo. Además se considera de gran importancia por la manera que desarrollan la investigación, y porque sirvió como guía para la propuesta de solución del presente trabajo.

En el análisis de las soluciones existentes se encontraron algunos trabajos similares, orientados al objetivo general de la investigación. Sin embargo, no se halló un artículo donde apliquen un método semiautomático para enriquecer una ontología geográfica a partir del procesamiento de base de datos espaciales, y que además sea visualizada en la web, brindando la posibilidad de enriquecerse aún más, con el aporte de un especialista mediante la edición de la misma. Por otro lado, mediante el estudio realizado a dichos trabajos ayudó a definir una mejor propuesta de solución para resolver la problemática planteada.

Conclusiones Parciales

Enfocado en el objetivo general de la investigación se realizó un estudio de los conceptos fundamentales relacionado con la ontología; se elige la definición propuesta por Hess para realizar el proceso de extracción de los componentes ontológicos de la base de datos espacial. Para el proceso de construcción de la ontología, se analizaron las herramientas y lenguajes fundamentales, y basado en las ventajas y desventajas de cada uno de ella se selecciona el *framework* Jena y el lenguaje OWL para el desarrollo del objetivo de la investigación. En el estudio realizado al estado del arte se encontraron varios trabajos que valieron para el desarrollo de la investigación; las soluciones existentes permitieron adentrarse en los temas que fueran de interés para desarrollar la solución propuesta.

CAPÍTULO 2 SELECCIÓN DE LAS TECNOLOGÍAS. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

Introducción

En el presente capítulo se describirá el proceso llevado a cabo durante la fase de análisis y diseño de solución siguiendo un enfoque teórico, donde se analiza la solución. Se fundamenta la selección de las herramientas a utilizar en cada una de las fases que componen la propuesta teórica, centrándose la misma en el procesamiento de lenguaje natural, y se emiten valoraciones sobre esas herramientas. Se define un modelo de dominio, se describen los requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema. También se establece la arquitectura de la solución.

2.1 Descripción de la solución que se propone

La solución propuesta está diseñada para aquellos usuarios que desean construir una ontología o trabajar sobre una ya existente, con el propósito de enriquecerla semánticamente, partiendo del procesamiento de cualquier base de datos espacial, garantizando generar, mantener y expandir con adecuada calidad ontologías de gran tamaño.

Se han encontrado trabajos similares donde utilizan diversas fuentes de información para la generación automática de ontologías (Fu, 2005); (Ping, y otros, 2009). Partiendo de que los SIG generalmente ya tienen una base de datos espacial, se escoge a la misma como fuente de información para la presente investigación.

Para definir la semántica de los datos que se encuentran almacenados en una Base de Datos Espacial, es necesario precisar una ontología del dominio geográfico, que conceptualice los fenómenos geográficos del mundo real en función del esquema de la BDE y la información correspondiente al dominio en el que es empleada la misma. Por lo antes planteado, se selecciona Kaab-Ontology como ontología del dominio geográfico a enriquecer en el método propuesto a continuación:

El método EDB-to-Geo-Ontology cuenta con cuatro pasos fundamentales:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Extracción de la Información: Basado en la definición de ontología planteada por Hess se procesa la base de datos espacial. Se tratarán a los conceptos como los nombres de las tablas, siendo este un fenómeno de interés a ser representado en la ontología. Las instancias serían las tuplas que sean la ocurrencia de los nombres de las tablas, es decir, un concepto. Las propiedades van a ser componentes asociados a un concepto, que pueden ser de tipo dato u objeto. En caso de que sea de tipo dato se trataría como *string*, *integer*, *double*, etc. como se conoce normalmente en programación o vista como atributos de base de datos, y las propiedades de tipo objeto puede ser vista como una relación de base de datos, donde se define como la relación que exista entre dos tablas.

Construcción: Definido en el primer paso de cómo se van a extraer los componentes ontológicos, automáticamente se construye la ontología.

Integración: Se elige una ontología del dominio geográfico (Kaab-Ontology) que será integrada con la ontología construida a partir del procesamiento de la base de datos, devolviendo una ontología enriquecida producto a la incorporación de nuevos componentes a la misma.

Edición: La ontología resultante producto a la integración de las ontologías, se visualizara en una herramienta que permita visualizar y gestionar todos los componentes ontológicos que contiene la misma, con el objetivo de perfeccionarla aún más.

2.2 Análisis de la Metodología

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, incremental, etc.) y definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas.

La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. La misma comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software, desde que surge la necesidad del producto hasta que se cumplió el objetivo por el cual fue creado. Una definición estándar de

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

metodología puede ser el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla (INTECO, 2009).

Actualmente la plataforma GeneSIG en su versión 2.0 define a PRODESOFIT como metodología de desarrollo de software a emplear por todos los productos que se desarrollan en el mismo. Por tal motivo y atendiendo a sus características, se selecciona la misma para el desarrollo de la herramienta de software, debido a que la integración del método propuesto está dirigido a esta plataforma.

Esta metodología está encaminada a satisfacer necesidades específicas del centro, apoyada en la combinación de varios modelos de desarrollo de software: el basado en componentes, el iterativo y el incremental.

Características del uso de Metodología de Desarrollo de Software:

- ✓ Optimiza el proceso y el producto software.
- ✓ Métodos que guían en la planificación y en el desarrollo del software.
- ✓ Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

El ciclo de vida proporciona el marco de referencia básico para dirigir el proyecto, independientemente del trabajo específico involucrado en cada fase. El mismo está compuesto por cinco fases: inicio, modelación, construcción, explotación experimental y despliegue. Cada fase terminará en un hito¹¹ con el objetivo fundamental de evaluar y decidir el paso a la siguiente fase de desarrollo.

¹¹ Un hito es una actividad de duración nula (0 horas). Los hitos se utilizan como marcadores o puntos de control. Indican fechas o puntos dentro del proyecto que se han de tener en especial consideración.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

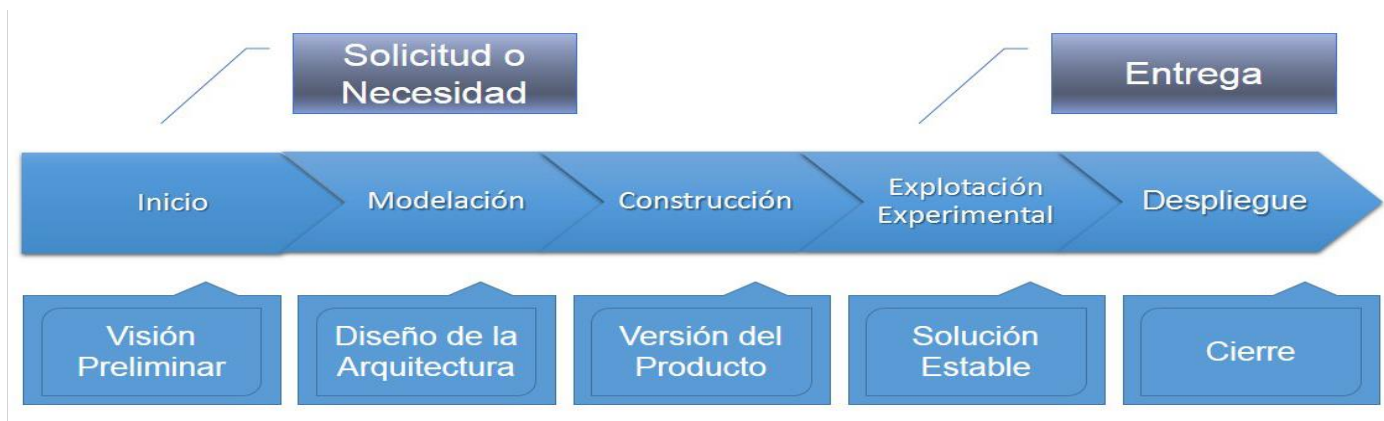


Figura 2 Etapas del Ciclo de Vida

2.3 Tendencias actuales en el desarrollo de sistemas informáticos

En el proceso desarrollo de software es fundamental la correcta elección de las tecnologías y herramientas que mejor se adapten y mayores facilidades brinden para dar solución al problema de la investigación.

2.2.4 Herramientas CASE (*ingeniería de software asistida por computadora*) para el modelado

En ingeniería de software se han establecido diversos procesos, metodologías y herramientas para estandarizar y facilitar el desarrollo de sus productos. Entre este grupo de herramientas se encuentran las CASE, las cuales soportan en forma automática varios o todos los pasos de dichas metodologías y se enmarcan en la ingeniería del software. Las herramientas CASE ayudan a reducir el tiempo empleado en el desarrollo de un sistema, lo que mantiene el costo estable y contribuye a mejorar su calidad, además permiten al analista documentar y modelar un sistema, desde la definición de requerimientos hasta el diseño, implementación y prueba (Maria, 1999).

Para alcanzar calidad en el producto final se definen un conjunto de objetivos (Maria, 1999):

- ✓ Automatizar la generación de documentación.
- ✓ Mejorar la calidad mediante la automatización de la comprobación de errores.
- ✓ Facilitar la compartición de información entre varios proyectos y la reutilización del software.
- ✓ Aprobar un entorno de desarrollo interactivo.
- ✓ Simplificar el lado del mantenimiento.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Estas herramientas constituyen uno de los factores relevantes para la correcta planificación, estimación, diseño, implementación y detección de errores de los proyectos productivos realizados en la Universidad de las Ciencias Informáticas por lo que los equipos de desarrollo utilizan este tipo de tecnologías con frecuencia para lograr una mejor productividad y calidad en los productos informáticos, así como la reducción del tiempo de construcción de los mismos. De esta forma se puede evidenciar la importancia que posee realizar una correcta selección de la herramienta CASE para el trabajo deseado, es por esto que a continuación se definen una serie de aspectos con los cuales la misma deberá de cumplir:

- ✓ Facilidad de uso.
- ✓ Modelar el software a través de diagramas UML.
- ✓ Lograr integración con diferentes IDE.
- ✓ Generar código en el lenguaje utilizado para el desarrollo.

Visual Paradigm for UML 8.0

Es una poderosa herramienta CASE que utiliza UML para el modelado; es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (Huanca, 2015).

Visual Paradigm ofrece (Venete, 2011)

- ✓ Un entorno de creación de diagramas para UML 2.0.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad de integrarse en los principales IDEs.
- ✓ Disponibilidad en múltiples plataformas.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Según las características planteadas anteriormente se decidió utilizar la herramienta CASE Visual Paradigm para UML pues la misma es la que más se ajusta a las necesidades del proyecto, es una herramienta profesional que soporta todo el ciclo de vida del software y permite el desarrollo de aplicaciones con rapidez, facilidad y calidad con un menor costo de producción.

2.2.3 Entorno de Desarrollo Integrado, IDE

Un Entorno de Desarrollo Integrado, traducido del inglés *Integrated Development Environment* o como su siglas en ingles lo indican IDE es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE puede denominarse como un entorno de programación que ha sido tratado como un programa aplicación. Esto significa que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Entre las características fundamentales que debe tener presente un IDE se encuentra:

- ✓ Multiplataforma.
- ✓ Soporte para diversos lenguajes de programación.
- ✓ Integración con sistemas de control de versiones.
- ✓ Reconocimiento de sintaxis.
- ✓ Extensiones y componentes para el IDE.
- ✓ Integración con *framework* populares.
- ✓ Depurador.
- ✓ Importar y exportar proyectos.

Netbeans 8.0

El IDE ¹² y Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de gran envergadura, tanto web como de escritorio. Distintas empresas especializadas en desarrollo de software, suministran extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. NetBeans admite la creación de aplicaciones Web en PHP, lo cual representa una ventaja para el desarrollo

¹² *Integrated Development Enviroment: Siglas en ingles de Entorno Integrado de Desarrollo.*

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

de GeneSIG, cuya estructura está programada del lado del servidor en PHP, y NetBeans además es capaz de reconocer las librerías Ext JS de JavaScript, con las que se han implementado las interfaces de usuario y las validaciones del lado del cliente. Además tiene soporte para AJAX donde agilizan la transmisión de datos en la Web. Este IDE es muy potente y además para el desarrollo de GeneSIG no necesita hacer uso de librerías específicas del mismo, lo cual representa una ventaja en la portabilidad del producto GeneSIG hacia otros IDEs. Además cuenta con la potencialidad de ser software de código abierto, siguiendo la política de desarrollo de GeneSIG.

2.3.1 Lenguajes de Programación

Los lenguajes de programación son un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Permiten la creación de software y están constituidos por un conjunto de símbolos y reglas sintácticas y semánticas que definen la estructura y el significado de los elementos y las expresiones de estos. Es utilizado para controlar el comportamiento físico y lógico de una máquina, permitiéndole a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, representando en forma simbólica y en manera de un texto los códigos que podrán ser leídos y escritos por una persona que no necesariamente posea muchos conocimientos sobre programación (Lenguajes de Programación, 2009).

Teniendo en cuenta que para el Enriquecimiento de la Ontología Geográfica se utiliza tecnología de escritorio y para el desarrollo de la gestión de la misma se utiliza tecnología web, se realiza un estudio de los lenguajes de programación Java, PHP y JavaScript. Estas facilitan el desarrollo del plugin para la extracción y visualización de la información de la Ontología.

Java 1.8

Lenguaje de programación de alto nivel, muy extendido en los últimos años, con el cual se pueden realizar prácticamente cualquier tipo de aplicación. La principal característica con la que cuenta es que es independiente de la plataforma sobre la cual trabaje, ya sea Windows, Linux o Mac OS X. Su portabilidad

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

entre dichas plataformas se logra a través de la Máquina Virtual de Java que actúa como enlace entre el Sistema Operativo y el programa de Java posibilitando que este sea perfectamente entendible. La programación en lenguaje Java permite el desarrollo de aplicaciones bajo la arquitectura cliente-servidor, dentro de lo que se encuentra la llamada computación distribuida.

Esta característica permite que en resultado de esta investigación se muestra implementación de una aplicación servidora en Java que maneje las librerías del razonador Jena y publique un servicio que será consumido desde otra aplicación web en PHP + JavaScript (Ext JS & AJAX), esta última ajustándose al diseño de la plataforma GeneSIG como herramienta para la misma. Se considera para esta investigación que la utilización de estas tecnologías permite la interacción usuario-aplicación en la solución de manera instructiva, logrando sencillez y fácil maniobrabilidad sobre la misma, además de que haciendo uso de las mismas se logra mejor integración con la interfaz de la plataforma GeneSIG.

PHP 5.0

PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML¹³ (PHP, 2014). Es un lenguaje de programación que puede hacer todo tipo de cosas: evaluar datos de formularios enviados desde un navegador, construir contenido web a medida para el navegador, hablar a una base de datos, e incluso enviar y recibir cookies¹⁴. Lo que distingue un PHP del lado del cliente como Javascript¹⁵ es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente (PHP, 2014).

Entre las características de PHP 5.0 están la inclusión de:

La visibilidad: La visibilidad de una propiedad o método se puede definir anteponiendo una de las palabras claves *public*, *protected* o *private* en la declaración. Los miembros de clases declarados como *public* pueden ser accedidos desde cualquier lado. Los miembros declarados como *protected*, sólo desde la misma clase,

¹³ Es el lenguaje que se emplea para el desarrollo de páginas web. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla.

¹⁴ Las cookies son un mecanismo por el que se almacenan datos en el navegador remoto y poder así monitorizar o identificar a usuarios que vuelven al sitio web.

¹⁵ Es un lenguaje de programación que se utiliza principalmente del lado del cliente permitiendo crear efectos atractivos y dinámicos en las páginas web.

desde las clases que hereden de ella y desde las clases *parent*. Aquellos miembros definidos como *private*, únicamente pueden ser accedidos desde la clase que los definió.

Interfaces: Las interfaces de objetos permiten crear el código con el cual especifican qué métodos deben ser implementados por una clase, sin tener que definir cómo estos métodos son manipulados. Las interfaces son definidas utilizando la palabra clave *interface*, de la misma forma que con clases estándar, pero sin métodos que tengan su contenido definido. Todos los métodos declarados en una interfaz deben ser públicos, ya que ésta es la naturaleza de una interfaz.

Clonación: Cuando se clona un objeto, PHP 5.0 llevará a cabo una copia superficial de las propiedades del objeto. Las propiedades que sean referencias a otras variables, mantendrán las referencias.

2.3.2 Marco de Trabajo

En el proceso de desarrollo de software un marco de trabajo, también definido como infraestructura digital o *framework*¹⁶, es una estructura conceptual y tecnológica de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Puede ser considerado como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Puede incluir lenguaje interpretado, bibliotecas, soporte a programas y otras herramientas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (Microsoft, 2014).

Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Existen *framework* de desarrollo en GNU/Linux como es el caso de Jena.

Jena 2.12.0

Jena es un framework Java para construir aplicaciones basadas en ontologías. Se desarrolló en HP Labs en el 2000, en 2009 HP cedió el proyecto a la fundación Apache que decidió adoptarlo en noviembre de 2010. Cuenta con un marco de trabajo desarrollado en java (y para java) de código abierto, para construir aplicaciones para la Web semántica. Proporciona un entorno de programación para RDF, SPARQL, RDFS

¹⁶ Marco de Trabajo.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

y OWL, incluyendo un motor de inferencia basado en reglas. Es un proyecto Open-Source y crece con la ayuda del programa de WS de "HP Labs". Además proporciona una API para extraer datos y escribir en grafos RDF. Los gráficos se representan como un "modelo" a la moda. Un modelo puede ser de origen con datos de archivos, bases de datos, direcciones URL o una combinación de éstos. Un modelo también se puede consultar a través de SPARQL. Jena proporciona soporte para OWL (Lenguaje de Ontologías Web).

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.

Asociados con Jena hay un gran número de proyectos interesantes:

- ✓ Jastor: generador de JavaBeans desde OWL.
- ✓ Micro Jena: Jena para dispositivos móviles.
- ✓ D2RQ: permite trabajar con bases de datos relacionales no RDF con grafos Jena RDF.
- ✓ Jena Grande: colección de utilidades y ejemplos para usar MapReduce, Pig y HBase para procesar datos en formato RDF.

2.4 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 (Rumbaugh, 2000).

A continuación se presenta el Modelo de Dominio donde se exponen los principales conceptos y roles del sistema en cuestión:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

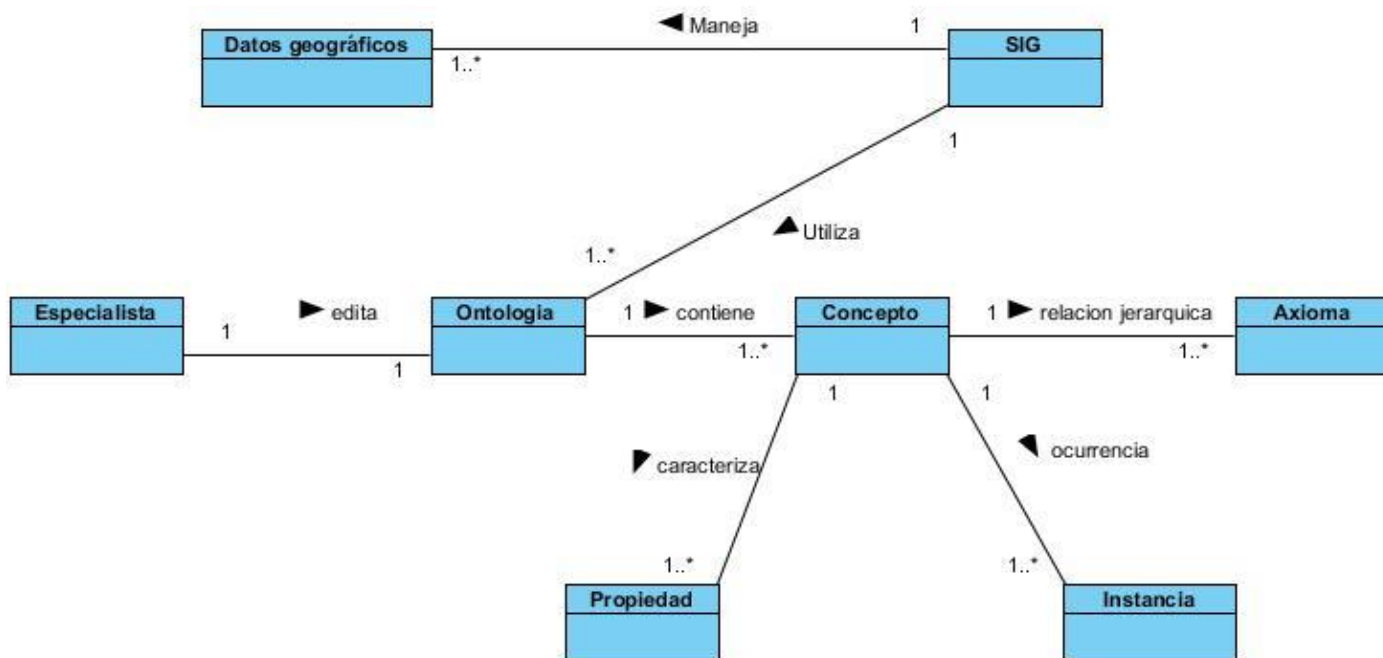


Figura 3 Modelo Conceptual

2.4.1 Descripción de los conceptos del modelo de dominio

- ✓ Especialista: Es el usuario que tiene los permisos de editar la ontología.
- ✓ Ontología: Está compuesta por conceptos, instancias, propiedades y axiomas.
- ✓ Concepto: Es cualquier fenómeno de interés a ser representado en la ontología.
- ✓ Instancia: Es una ocurrencia de un concepto.
- ✓ Propiedad: Es una componente que es asociada a un concepto con el objetivo de caracterizarlo.
- ✓ Axioma: Es la relación jerárquica entre los conceptos.
- ✓ Datos geográficos: Son los datos que maneja el SIG.
- ✓ SIG: Sistema de Información Geográfica.

2.5 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.

2.5.1 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 (Larman, 2004).

A continuación se abordan los requerimientos definidos para el desarrollo del software:

Proceso 1: Realizar análisis sobre los conceptos.

RF1: Extraer concepto de la base de datos espacial.

RF2: Insertar concepto a la ontología.

RF3: Modificar concepto a la ontología.

RF4: Eliminar concepto a la ontología.

RF5: Mostrar concepto de la ontología.

Proceso 2: Realizar análisis sobre las propiedades.

RF6: Extraer propiedad de la base de datos espacial.

RF7: Insertar propiedad a la ontología.

RF8: Modificar propiedad a la ontología.

RF9: Eliminar propiedad a la ontología.

RF10: Mostrar propiedad de la ontología.

Proceso 3: Realizar análisis sobre las instancias.

RF11: Extraer instancia de la base de datos espacial.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

RF12: Insertar instancia a la ontología.

RF13: Modificar instancia a la ontología.

RF14: Eliminar instancia a la ontología.

RF15: Mostrar instancia de la ontología.

Proceso 4: Realizar análisis sobre los axiomas.

RF16: Extraer axiomas de la base de datos espacial.

RF17: Insertar axiomas a la ontología.

RF18: Modificar axiomas en la ontología.

RF19: Eliminar axiomas en la ontología.

RF20: Mostrar axiomas.

Proceso 5: Realizar análisis sobre la ontología.

RF21: Construir ontología.

RF22: Enriquecimiento de la ontología del dominio geográfico.

RF23: Escribir fichero.

A continuación se muestra las especificaciones de los requisitos funcionales críticos de la solución:

Tabla # 1 Especificación del RF 21: Construir Ontología

Conceptos Tratados	Conceptos	Atributos
	Ontología	-

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

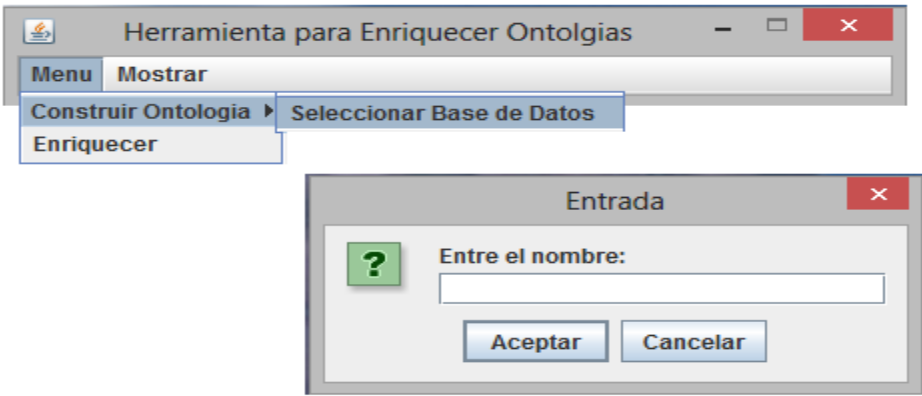
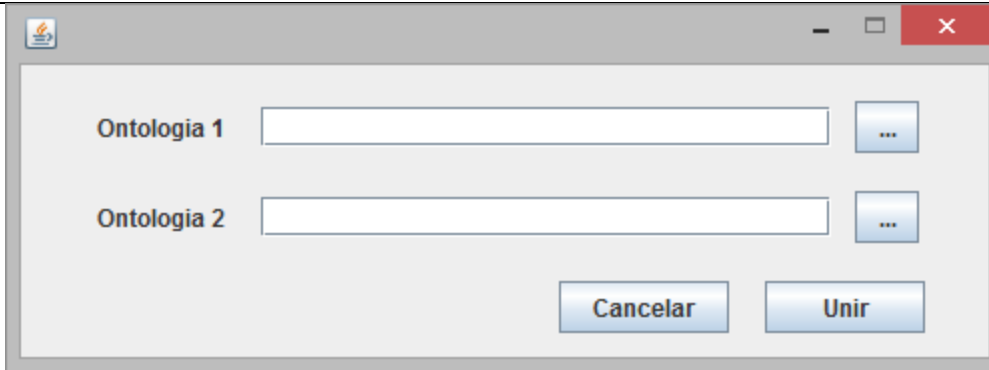
Precondiciones	Precondiciones	Pre-requisito
	Se debe haber extraído los componentes ontológicos.	RFs 1, 2, 6, 7, 11, 12, 16 y 17
Descripción	<ol style="list-style-type: none"> 1. Se entra el nombre de la base de datos que se procesará. 2. Se extraen los componentes ontológicos. 3. Se construye la ontología. 4. Se crea el fichero. 	
Validaciones	No se construirá la ontología si el nombre de la base de dato no se encuentra en el servidor.	
Post-condiciones	Se agrega los componentes a la ontología.	
Post-requisito	RFs 3, 4, 5, 8, 9, 10, 13, 14, 15, 18, 19, 20 y 23.	
Prototipo de interfaz		

Tabla # 2 Especificación del RF 22: Enriquecimiento de la ontología del dominio geográfico

Conceptos Tratados	Conceptos	Atributos
	Ontología	-
Precondiciones	Precondiciones	Pre-requisito
	Se debe haber construido la ontología.	RF 21

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Descripción	<ol style="list-style-type: none">1. Se entran las direcciones donde se encuentran las ontologías.2. Se integran.3. Se escribe el fichero.
Validaciones	No se unirán, si no entran las direcciones o deja algún campo vacío.
Post-condiciones	Se integran las ontologías.
Post-requisito	RF 23.
Prototipo de interfaz	

2.5.2 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, 2004), 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

- ✓ El sistema deberá ser utilizado por especialistas que posean al menos conocimientos básicos sobre el lenguaje de consultas a ontología.
- ✓ Debe contar con una interfaz amigable, con los colores básicos de la interfaz de la plataforma GeneSIG.

RNF2: Soporte

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

- ✓ Se utilizará el servidor web Apache para el trabajo con PHP y se utilizará PostgreSQL como sistema gestor de Base de Datos.
- ✓ El software contará antes de su puesta en marcha con un período de pruebas.

RNF3: Software

Para el cliente:

- ✓ Navegador Mozilla Firefox 3.6 o superior.
- ✓ Sistema operativo Windows XP o GNU/Linux.

Para el servidor:

- ✓ Sistema operativo GNU/Linux en cualquiera de sus distribuciones.
- ✓ Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible.
- ✓ Un servidor de base de datos PostgreSQL 9.4 o superior con su extensión “postgis” para datos espaciales.

RNF4: Hardware

Para el servidor de aplicaciones:

- ✓ Requerimientos mínimos: Procesador Pentium IV a 2GHz de velocidad de procesamiento y 2Gb de memoria RAM.
- ✓ Para la implementación se utilizará IDE NetBeans y la librería de Javascript ExtJS.
- ✓ Al menos 50Gb de espacio libre en disco duro.
- ✓ Tarjeta de red.

Para el servidor de base de datos:

- ✓ Requerimientos mínimos: Procesador Pentium IV a 2GHz de velocidad de procesamiento y 4Gb de memoria RAM.
- ✓ Al menos 50Gb de espacio libre en disco duro.
- ✓ Tarjeta de red.

Para el cliente:

- ✓ Requerimientos mínimos: Procesador Pentium IV a 1.2 GHz con 256Mb de memoria RAM.
- ✓ Tarjeta de red

2.6 Arquitectura de software

El concepto de arquitectura de software se refiere a la estructuración del sistema que se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo. El no crear este diseño desde etapas tempranas del desarrollo puede limitar severamente el que el producto final satisfaga las necesidades de los clientes. Además, el costo de las correcciones relacionadas con problemas en la arquitectura es muy elevado. Es así que la arquitectura de software juega un papel fundamental dentro del desarrollo (Cervantes, 2010).

De acuerdo con Pressman en el 2005, “se refiere a la estructura general del software y las maneras en que la estructura proporciona una integridad conceptual para un sistema. En su forma más simple, la arquitectura es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan, y la estructura de datos que utilizan los componentes. En un sentido más amplio, sin embargo, los componentes pueden generalizarse para representar elementos importantes del sistema y sus interacciones”.

2.6.1 Patrones arquitectónicos

Para el enriquecimiento de la ontología geográfica es fundamental el tratamiento que se les da a la información extraída de la base de datos espacial. Este proceso es completamente diferente al que se realiza en el plugin para GeneSIG, ya que en el mismo solo el usuario podrá visualizar la ontología y editarla. Al complementar estos procesos se logran enriquecer aún más la ontología, por esta razón se decide utilizar la arquitectura basada en componentes.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Teniendo en cuenta que un componente es un bloque de construcción modular para el software de cómputo, en otras palabras es una parte modular desplegable y reemplazable de un sistema que encapsula implementación y expone un conjunto de interfaces. Los componentes pueblan la arquitectura del software, ayudando a cumplir con los objetivos y requisitos del sistema en construcción (Pressman, 5ta Edición).

A continuación se muestra las características que identifican la arquitectura basada en componentes:

- ✓ Es un estilo de diseño para aplicaciones compuestas de componentes individuales.
- ✓ Pone énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen interfaces bien definidas.
- ✓ Define una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades.

Los principales beneficios del estilo de arquitectura basado en componentes:

Facilidad de desarrollo: Los componentes implementan una interfaz bien definida para proveer la funcionalidad definida permitiendo el desarrollo sin impactar otras partes del sistema.

Reusable: El uso de componentes reutilizables significa que ellos pueden ser usados para distribuir el desarrollo y el mantenimiento entre múltiples aplicaciones y sistemas.

Mitigación de complejidad técnica: Los componentes mitigan la complejidad por medio del uso de contenedores de componentes y sus servicios.

2.7 Modelo de Diseño

El principal propósito del modelo de diseño consiste en modelar el sistema y encontrar su forma para que soporte todos los requisitos funcionales así como para profundizar en los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con lenguajes de programación, sistemas operativos, tecnologías de interfaz de usuario y componentes reutilizables. Este modelo está compuesto por los diagramas de clases del diseño, los diagramas de interacción (secuencia y colaboración) aspectos dinámicos del sistema (Rumbaugh, 2000).

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

2.7.1 Diagrama de Clase del Diseño

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, se utilizan para modelar la vista de diseño estática de un sistema, y además se obtiene como resultado del refinamiento del modelo conceptual (Rumbaugh, 2000). Se muestran a continuación los diagramas de clases:

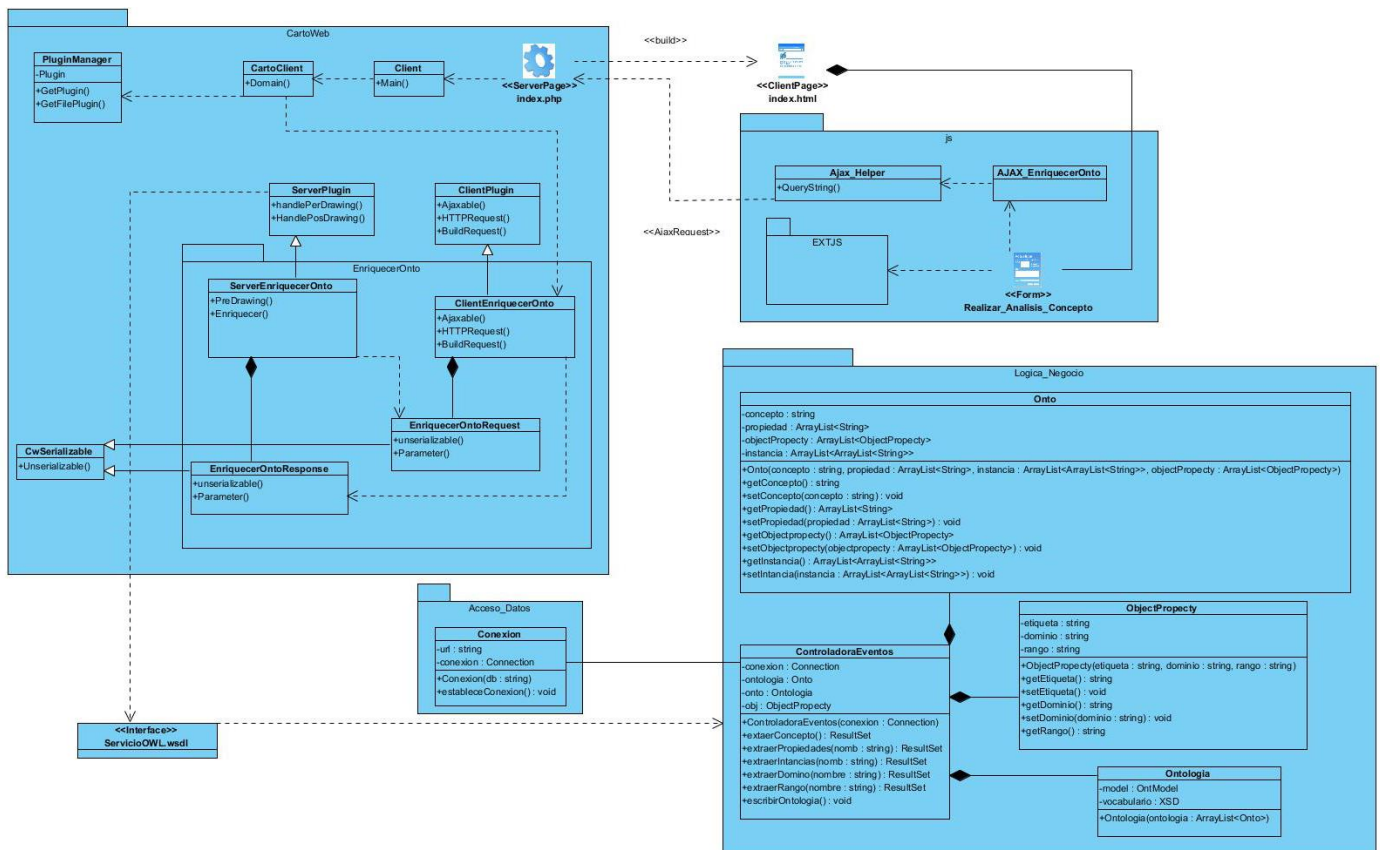


Figura 4 Diagrama de Clases del diseño del proceso 1. Realizar análisis sobre los conceptos.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

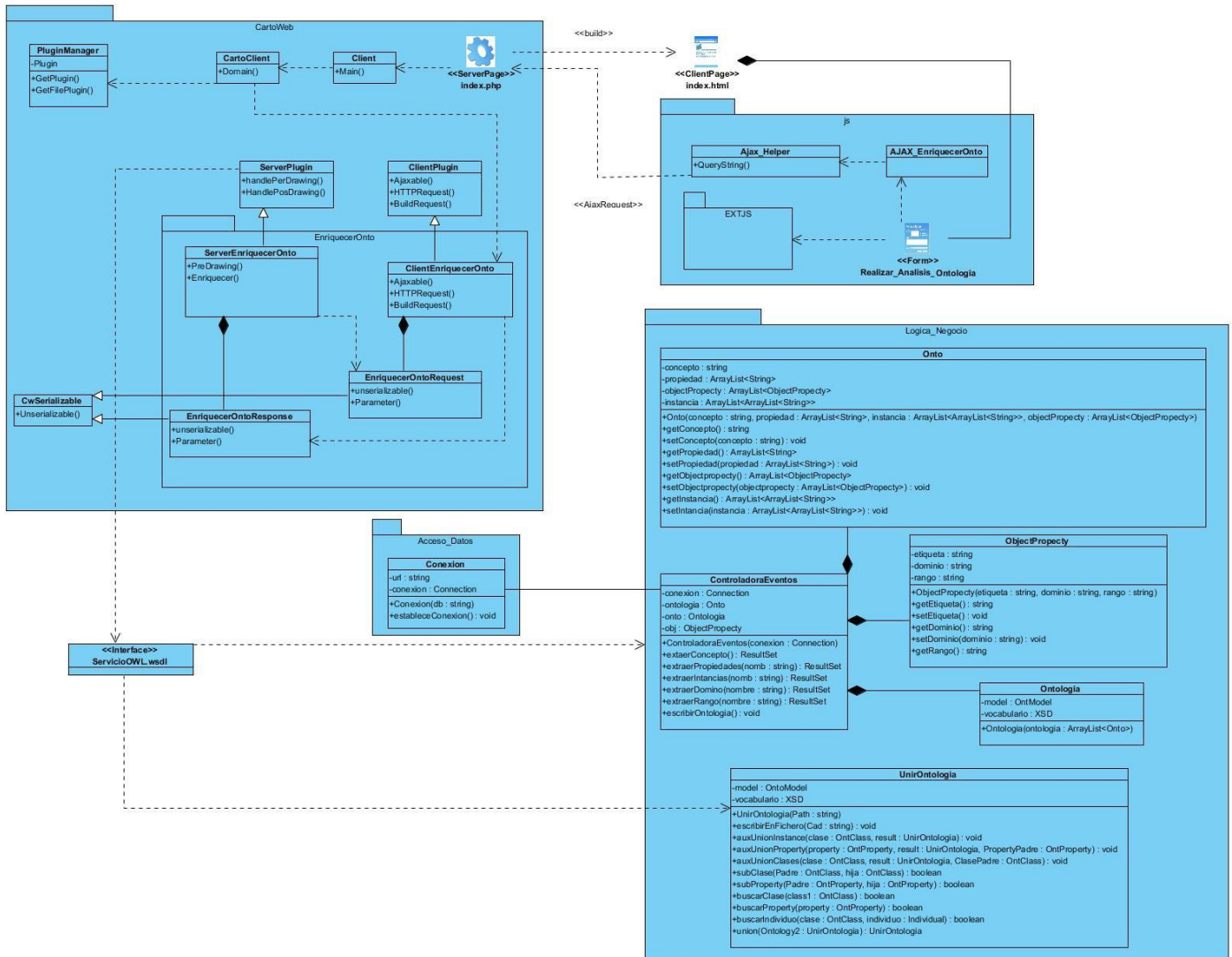


Figura 5 Diagrama de Clases del diseño del proceso 5. Realizar análisis sobre la ontología.

Descripción de las clases del diseño

Cada uno de los diagramas de clases de diseño que emplea el paquete CartoWeb posee clases que son comunes, así como también las clases que son JavaScript y PHP. Estas clases serán descritas a continuación:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

index.php: tiene como propósito controlar la realización del RF en sí, recibe las peticiones realizadas por el cliente, gestiona las mismas y manda a construir la ClientPage.

Client: contiene todos los archivos específicos de PHP del lado de CartoClient y permite la interacción entre la index.php y la CartoClient.

CartoClient: integra y recoge todos los datos y funciones realizadas por cada una de las .js que intervienen en el RF, y se definen una serie de variables globales que van a ser utilizadas por la aplicación.

PluginManager: clase que se utiliza para gestionar la base de plugins.

ClientPlugin: contiene las interfaces necesarias para los plugins del lado del cliente.

ServerPlugin: esta clase proporciona la base de herramientas para el desarrollo de plugins.

ServerContex: es la contenedora de la información común que ha de ser utilizada por la parte cliente y la servidora, empleando la información seleccionada como un objeto para un fácil manejo de los datos.

CwSerializable: se encarga de serializar todas aquellas clases que pueden ser serializadas, permitiendo la comunicación entre el Client y el Server del plugin.

AJAX_Helper: tiene como propósito enviar las respuestas de los plugins “AJAX”, para alimentar a los plugins que responden a las peticiones del usuario.

Index.html: es la encargada de mostrar en el mapa la región localizada.

AJAX_EnriquecerOnto: es la encargada de gestionar el pedido y respuesta a las peticiones del usuario por Ajax.

Realizar_Analisis_Ontologia: tiene como objetivo dado el nombre de una base de datos, construir una ontología, luego integrarla con una del dominio geográfico, y exportar un fichero donde se encuentra almacenada la ontología resultante de la unión.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

ServerEnriquecerOnto: es la clase servidora que tiene como principal función la conexión con la base de datos y con el paquete que propicia el trabajo con la ontología para realizar las consultas requeridas y enviar las respuestas necesarias al **ClientEnriquecerOnto**.

ClientEnriquecerOnto: se encarga de recoger y seleccionar de las .js contenidas en el paquete JS, toda la información correspondiente a los datos a precisar, entrados a través de los formularios, y los envía al **ServerEnriquecerOnto**.

EnriquecerOntoRequest: es una clase común encargada de transportar los datos recogidos en **ClientEnriquecerOnto** desde la interfaz y transportarlos a la clase **ServerEnriquecerOnto**.

EnriquecerOntoResult: es una clase común encargada de transportar los datos generados en **ServerEnriquecerOnto** a la clase **ClientEnriquecerOnto**.

ControladoraEventos: es la clase controladora, que se encarga de realizar todo el proceso de extracción de la información, construcción de la ontología y escribir el fichero.

Onto: es usada para el manejo de las ontologías.

ObjectProperty: es usada para el manejo de las propiedades de objeto.

Ontologia: es la encargada de agregar los componentes a la ontología para crear el fichero.

UnirOntologia: es la clase que se encarga de unir las ontologías.

2.7.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y para lograr una mayor calidad en el diseño (Bass, y otros, 2003). Estos patrones tienen su origen en el diseño de sistemas con el paradigma de la Programación Orientada a Objetos y en problemas que estos no pueden resolver y que son muy recurrentes en la programación de software. Los patrones de diseño ayudan a mantener un código reutilizable y tener mayor control sobre los problemas que son recurrentes, de ahí que sean una solución factible para su implementación en múltiples aplicaciones (Microsoft, 2014).

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Entre los patrones de diseño se encuentran dos importantes grupos: GRASP y GOF. Los patrones GRASP son los encargados de guiar los principios fundamentales de la asignación de responsabilidades en el diseño orientado a objeto y los patrones GOF los cuales se clasifican en tres grupos fundamentales: los patrones Creacionales que es donde se inicializan y configuran los objetos, los patrones Estructurales donde se separa la interfaz de la implementación y los patrones de Comportamiento los cuales describen el comportamiento entre las clases y los objetos.

Durante el desarrollo del presente trabajo de diploma se hace uso de los siguientes patrones:

Creador: este patrón es quien guía el proceso de asignación de responsabilidades relacionadas con la creación de objetos. Tiene el objetivo de asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este patrón se pone de manifiesto en la clase **ControladoraEventos**.

```
Onto ontologia = new Onto(concepto, propiedad, instancia, objectpropecty);
```

Figura 6 Patrón Creador

Experto: es el encargado de asignar una responsabilidad al experto en información, este tipo de patrón es utilizado a la hora de determinar las responsabilidades con las cuales debe cumplir cada clase, el mismo se evidencia en las clases **ObjectPropecty**, **Ontologia**, **Onto** y **UnirOntologia**.

Controlador: esta es el responsable de asignar a clases específicas la responsabilidad de controlar la salida de sucesos en el sistema, permitiendo a través de esta interactuar con las demás clases del componente. Este patrón se ve reflejado en la clase **ControladoraEventos**.

Singleton: consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Se evidencia en la clase **ControladoraEventos**, donde se crea la instancia para usarse en los métodos `extraerConcepto`, `extraerPropiedades`, `extraerInstancias`, `extraerDominio` y `extraerRango`.

Conclusiones Parciales

El estudio de las soluciones para enriquecer una Ontología Geográfica a partir de la extracción de la información de una base de datos espacial, permitió analizar las funcionalidades que pueden servir en el desarrollo de la solución, así como la selección de los principales componentes tecnológicos y herramientas a utilizar en el desarrollo de la solución propuesta, tanto para el diseño como en la implementación. Teniendo en cuenta que la plataforma GeneSIG se encuentra en desarrollo bajo tecnologías señaladas, se seleccionó la metodología PRODESOFIT 1.5, el lenguaje de programación PHP 5.0, entorno de desarrollo integrado NetBeans 8.0 y las herramientas Visual Paradigm for UML 8.0 para el modelado. Para la extracción de la información el lenguaje de programación Java 1.8, haciendo uso de Jena como framework de desarrollo; y como sistema de gestión de base de datos PostgreSQL 9.4, siendo el PgAdminIII 1.16.1 el gestor de PostgreSQL. A través del modelo de dominio queda reflejada las principales actividades que se realizan, se expusieron los requisitos que debe tener el software y se explica el diseño, tanto lógico como físico de la arquitectura de la solución, lo que permitió fijar las bases para el siguiente flujo de trabajo: la implementación.

CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

En este capítulo se brindan los aspectos referentes a la implementación y se realiza la validación de la solución propuesta en el capítulo anterior. Las pruebas son el proceso de ejecución de una aplicación con el propósito de comprobar que el producto satisfaga los requerimientos expresados y que el mismo tenga el comportamiento esperado. Para que las pruebas sean exitosas, se hace necesario confeccionar casos de pruebas que tengan probabilidades de descubrir los errores del sistema utilizando técnicas que rijan el proceso de la prueba.

3.1 Diagrama de componentes

Un componente es una parte del sistema sustituible, casi independiente e importante que desempeña una función clara en el contexto de una arquitectura bien definida (Sommerville, 2004). Los diagramas de componentes visualizan la relación física existente entre los componentes considerados necesarios para la programación de los casos de usos ya definidos, además son utilizados para estructurar y controlar los componentes en los sistemas de software, fundamentalmente mostrando las dependencias existentes entre estos.

A continuación se muestra el diagrama de componente:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

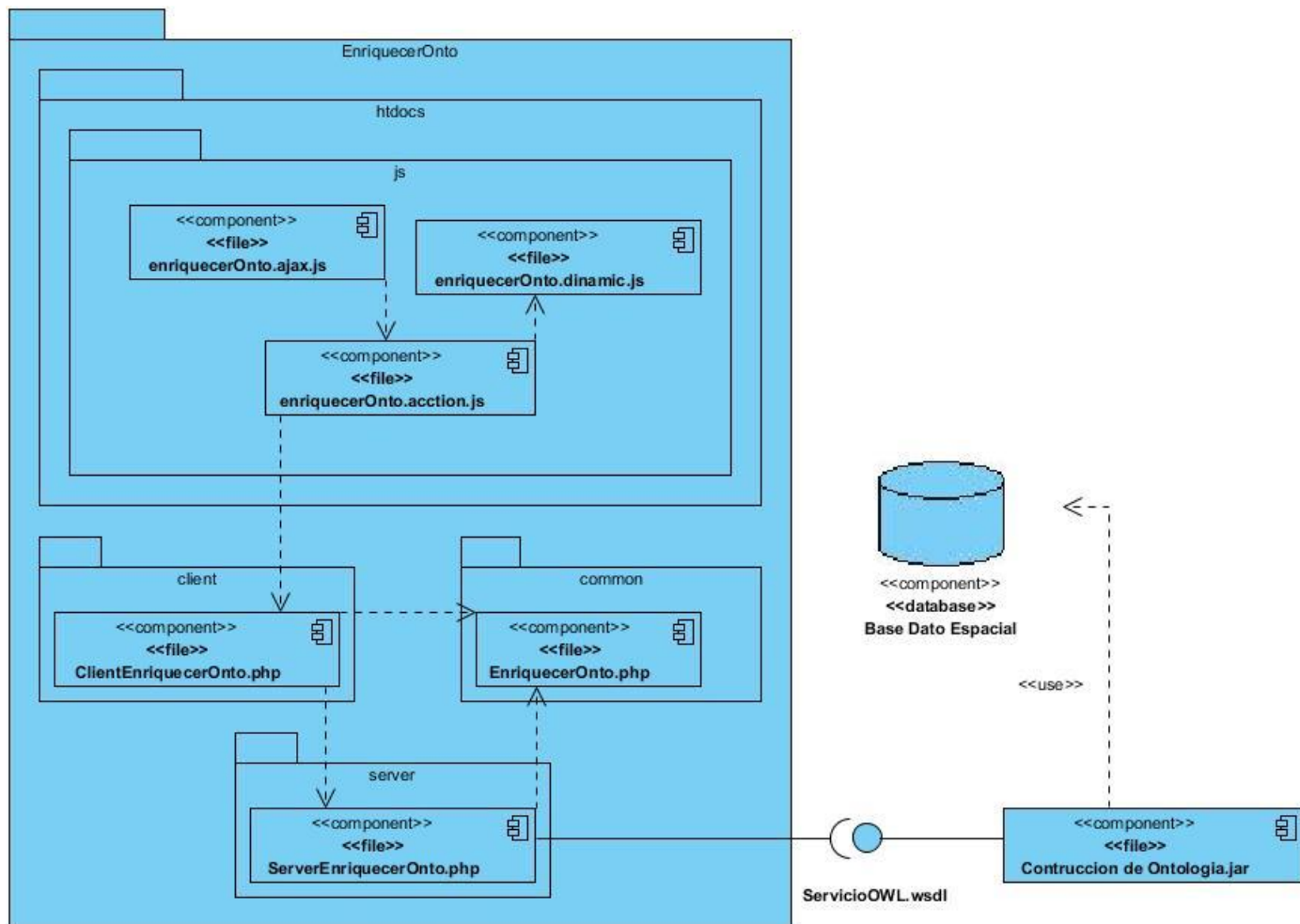


Figura 7 Diagrama de Componente

3.2 Modelo de Despliegue

El modelo de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Los nodos representan objetos físicos existentes en tiempo de ejecución, sirven para modelar recursos que tiene memoria y capacidad de proceso y puede ser tanto ordenadores como dispositivos, memoria o personas (Melchor, 2012). A continuación se representa el modelo de despliegue:

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

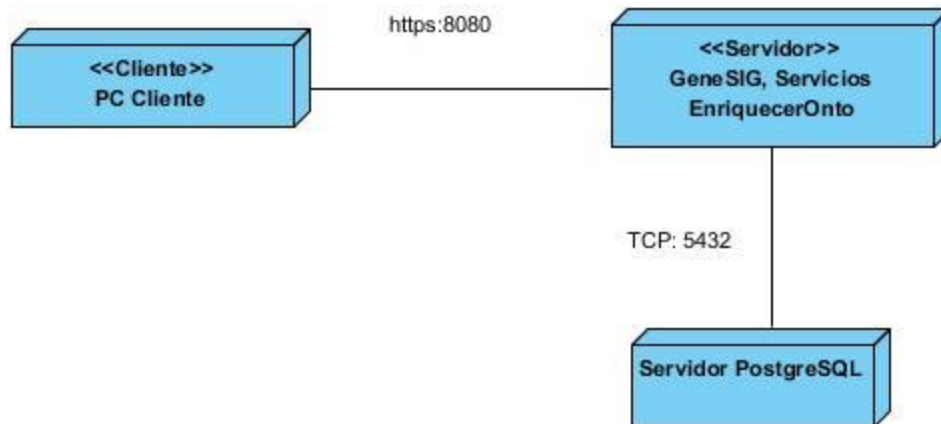


Figura 8 Modelo de Despliegue

3.3 Estándares de Codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, establezca un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (Microsoft, 2015).

Con la definición de los estándares de codificación se garantiza un estilo de programación homogéneo todos los participantes comprendan mejor el producto en menor tiempo y resulte fácil el futuro mantenimiento del mismo. Mantener un estándar de codificación es importante en la programación, reduce la posibilidad de cometer errores y como consecuencia obtener un producto con buena calidad.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Para el desarrollo de la herramienta de software se utilizaron los estándares de codificación UpperCamelCase¹⁷ para la representación de los nombres de las clases, y lowerCamelCase¹⁸ para los nombres de los métodos, los mismos se muestran a continuación:

```
public class ControladoraEventos {  
  
    private Connection conexion;  
  
    public ControladoraEventos(Connection conexion) { ...3 lines }  
  
    public ResultSet extraerConcepto() { ...11 lines }
```

Figura 9 Estándares de Codificación

3.4 Pruebas del software

Una de las etapas en el desarrollo del software son las pruebas, las cuales son los procesos que permiten validar y revelar la calidad de un software, básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema (Tenorio, 2010).

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Las pruebas constituyen el último bastión desde el que se puede evaluar la calidad y, de forma más pragmática, descubrir los errores. Pero las pruebas no deben ser vistas como una red de seguridad (Pressman, 5ta Edición).

El único instrumento adecuado para determinar la calidad de un producto de software, es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de

¹⁷ UpperCamelCase: Cuando la primera letra de una frase es mayúscula. Ejemplo: #UnSencilloEjemplo.

¹⁸ LowerCamelCase: Varía con relación a la anterior en que la primera letra de la frase es minúscula. Ejemplo: #unSencilloEjemplo.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. El objetivo fundamental por el cual se realizan las pruebas es (Barrios, 2014):

- ✓ La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- ✓ Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- ✓ Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Con la intención de garantizar la calidad requerida, el cumplimiento de los requisitos especificados y facilitar el enriquecimiento de la ontología geográfica a partir de una base de datos espacial, se desarrollaron pruebas que garantizan el cumplimiento de dichas acciones.

3.4.1 Tipos de Pruebas

Las pruebas pueden ejecutarse en cualquier punto del proceso de desarrollo de software, y son capaces de proporcionar al sistema desarrollado la calidad requerida por el cliente. Existen las Pruebas Unitarias que tienen ese nombre debido a que se prueba la funcionalidad de cada método o función; únicamente contemplando la lógica que debe realizar y excluyendo la convivencia con otras clases o sistemas. Las Pruebas Funcionales son pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Otra de las pruebas que garantizarían la calidad del sistema es la Prueba de Integración, estas son similares a las funcionales, pero sobre repositorios reales, datos reales y con la interacción real con otros sistemas o componentes (Barrios, 2014).

La Prueba de Aceptación de Usuario es ejecutada antes de que la aplicación sea instalada dentro de un ambiente de producción. La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar como el sistema satisface sus criterios de aceptación validando los requisitos que han sido levantados para el desarrollo, incluyendo la documentación y los procesos de negocio. Las Pruebas de Integración son ejecutadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta (Zapata, 2013).

3.5 Prueba Realizadas

Prueba de Caja Blanca

Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones. Ejemplos típicos de ello son las pruebas unitarias. Se centran en lo que hay codificado o diseñado a bajo nivel por lo que no es necesario conocer la especificación de requisitos, que por otra parte será difícil de relacionar con partes diseñadas a muy bajo nivel (Carabali, 2013).

El método del camino básico, técnica de prueba de Caja Blanca, permite obtener una medida de la complejidad lógica del diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa (Pressman, 5ta Edición).

Para aplicar correctamente esta prueba de Caja Blanca es preciso seguir los siguientes pasos:

1. Generar grafo de flujos de datos.
2. Cálculo de complejidad ciclomática, $V(G)$.

$$V(G) = NA \text{ (Número de Aristas)} - NN \text{ (Número de Nodos)} + 2.$$

$$V(G) = P \text{ (Nodos predicados)} + 1.$$

$$V(G) = \text{Número de regiones.}$$

3. Determinar los caminos independientes o básicos.
4. Generar un caso de prueba para cada camino de ejecución.

Con la utilización de este método de camino básico es posible la detección de errores no detectados anteriormente en el código. De igual manera el mantenimiento del sistema sería de forma más efectiva y certera.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

```
public void escribirOntologia() throws SQLException, IOException {
    ResultSet rs = extraerConcepto();
    ArrayList<String> propiedad = new ArrayList<String>();
    ArrayList<String> tipoprop = new ArrayList<String>();
    ArrayList<ArrayList<String>> intancia = new ArrayList<ArrayList<String>>();
    try {
        while (rs.next()) {
            String concepto = rs.getString(1);
            ResultSet propiedades = this.ExtraerPropiedades(concepto);
            System.out.println(ExtraerPropiedades(concepto));
            while (propiedades.next()) {
                propiedad.add(propiedades.getString(1));
                System.out.println(propiedades.getString(1));
            }
            ResultSet intancias = ExtraerIntancias(concepto);
            while (intancias.next()) {
                int row = propiedad.size();
                ArrayList<String> arreglo = new ArrayList<String>();
                for (int i = 1; i <= row; i++) {
                    arreglo.add(intancias.getString(i));
                }
                intancia.add(arreglo);
            }
            Ontologia onto = new Ontologia(concepto, propiedad, intancia);
            propiedad = new ArrayList<String>();
            intancia = new ArrayList<ArrayList<String>>();
        }
    } catch (Exception e) {
        System.out.println("Error");
    }
}
```

Figura 10 Método escribirOntologia de la clase ControladoraEventos

Paso # 1: Generar el grafo del método que se muestra en la Figura 16.

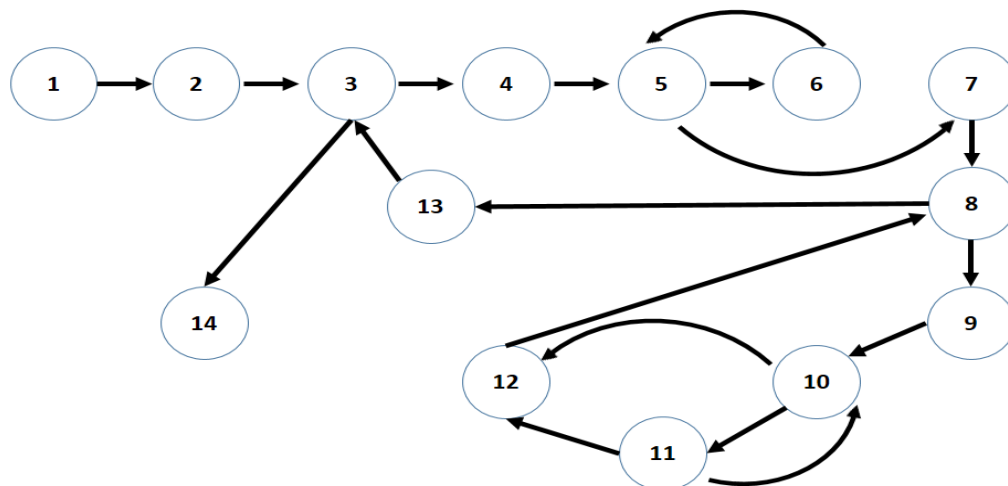


Figura 11 Grafo que genera el método

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Paso # 2: Calcular la complejidad.

$$V(G) = NA \text{ (Número de Aristas)} - ND \text{ (Número de Nodos)} + 2$$

$$V(G) = 18 - 14 + 2$$

$$V(G) = 6$$

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = 5 + 1$$

$$V(G) = 6$$

$$V(G) = R \text{ (Número de regiones)}$$

$$V(G) = 6$$

Paso # 3: Determinar los caminos básicos.

CB1: 1-2-3-14

CB2: 1-2-3-4-5-7-8-13-3-14

CB3: 1-2-3-4-5-6-5-7-8-13-3-14

CB4: 1-2-3-4-5-7-8-9-10-12-8-13-3-14

CB5: 1-2-3-4-5-7-8-9-10-11-10-12-8-13-3-14

CB6: 1-2-3-4-5-6-5-7-8-9-10-12-8-13-3-14

Paso # 4: Caso de prueba para el camino básico. En este paso se le harán las pruebas a 3 caminos, considerados los más importantes de los 6 caminos básicos obtenidos, debido a que el método la función

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

que ejerce, es de escribir sobre la ontología partiendo del concepto inicial, por tanto las pruebas a todos los caminos sería innecesario ya que la lógica es la misma.

Caso de prueba para CB1:

Entrada: rs.next

Resultado esperado: Adicionar todas las propiedades y las instancias en un arreglo para escribir en la ontología.

Resultado de la prueba: Satisfactorio.

Caso de prueba para CB2:

Entrada: propiedades.next

Resultado esperado: Escribir todas las propiedades.

Resultado de la prueba: Satisfactorio.

Caso de prueba para CB1:

Entrada: instancias.next

Resultado esperado: Escribir todas las instancias.

Resultado de la prueba: Satisfactorio.

Se realizó la este tipo de prueba para garantizar, entre otros aspectos, que se ejerciten por lo menos una vez todos los caminos independientes de cada método. Es por ello que la prueba de Caja Blanca es uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Prueba de caja Negra

Para garantizar el correcto funcionamiento de la interfaz del software se realizan pruebas funcionales de Caja Negra. Se centra en los requisitos funcionales, que se aplican sobre la herramienta de software sin necesidad de conocer como está construido por dentro. Las mismas se aplican a un determinado conjunto de datos de entrada y observando las salidas que se producen para identificar si la función se está desempeñando correctamente en la interfaz a prueba. Se observa la funcionalidad y debe contrastar con la especificación que se le asigna (Carabali, 2013).



Figura 12 Funcionamiento de la Prueba de Caja Negra

En esta fase de prueba se realizaron tres iteraciones. Como parte de la primera iteración se identificó una no conformidad en el requisito funcional 16: Construir ontología, el mismo presentaba errores que no garantizaban el correcto funcionamiento del mismo. En la segunda iteración se identifica una no conformidad en el requisito funcional 17: Integrar ontología, el mismo presentaba problemas en el proceso de integración. La tercera iteración fue realizada con el objetivo de verificar si había sido erradicado las no conformidades antes encontradas y comprobar si el resto de los requisitos funcionaban correctamente. A continuación se muestra los casos de prueba resultantes en la tercera iteración realizado a los requisitos funcionales más importantes.

Tabla # 3 Secciones a probar en el RF16: Construir ontología.

Nombre de la sección	Escenario	Acción realizada	Respuesta del software	Resultados de la prueba
SC 1: Construir ontología.	EC 1.1: Generar la ontología.	El usuario introduce el nombre de la base de datos.	El software valida que los datos estén correctos y construye la ontología.	Resultado satisfactorio.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

	EC 1.2: Nombre incorrecto.	El usuario entra un nombre de base de datos que no existe.	Se muestra un mensaje que alerta sobre la no existencia de la base de datos. El software cancela la operación realizada por el usuario.	Resultado satisfactorio.
--	-----------------------------------	--	---	--------------------------

Tabla # 4 Secciones a probar en el RF17: Integrar ontología.

Nombre de la sección	Escenario	Acción realizada	Respuesta del software	Resultados de la prueba
SC 1: Integrar ontología.	EC 1.1: Unir ontología.	El usuario entra las direcciones donde se encuentran las ontologías.	El software valida que se hayan entrado las direcciones e integra las ontologías.	Resultado satisfactorio.
	EC 1.2: Dirección incorrecta.	El usuario no ingresa las direcciones donde se encuentran las ontologías.	Se muestra un mensaje que alerta la existencia de campos vacíos o dirección errónea. El software cancela la operación realizada por el usuario.	Resultado satisfactorio.

Pruebas de aceptación

Una prueba de aceptación es un escenario de utilización del sistema y el comportamiento que de él se espera, visto desde la perspectiva del cliente, usuario o sistema externo que interactúa con el programa. Se define las Pruebas de Aceptación (PA) como: “el propósito de demostrar al cliente el cumplimiento de un requisito del software”.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Características de una PA son:

- ✓ Describe un escenario (secuencia de pasos) de ejecución o uso del sistema desde la perspectiva del cliente. Puede estar asociada a un requisito funcional o requisito no funcional.
- ✓ Un requisito tiene una o más PAs asociadas.
- ✓ Las PAs cubren desde escenarios típicos/frecuentes hasta los más excepcionales.
- ✓ Una PA puede tener infinitas instanciaciones (ejecuciones con valores concretos). El diseño de las instanciaciones y su aplicación es trabajo del tester¹⁹.

Las pruebas de aceptación constituyen el criterio de éxito en cuanto a la implementación de un requisito del sistema. Un mismo requisito del sistema puede presentarse en ejecución como diferentes escenarios (por ejemplo alternativas correspondientes a elecciones que realiza el usuario al interactuar con el sistema). La idea es que junto con la definición de la unidad de trabajo se definan las pruebas de aceptación.

Las pruebas de aceptación deben ser validadas con el cliente y verificadas (en cuanto a corrección y completitud) por otros miembros del equipo asignados a la unidad de trabajo. Es importante que el analista establezca un orden en las pruebas, desde las correspondientes a escenarios más típicos/frecuentes a aquellos más excepcionales. Éste orden servirá de guía de implementación a los programadores y al trabajo del tester. El programador debería escribir el código con la idea de superar las pruebas de aceptación de forma incremental, hasta implementar toda la funcionalidad y pasar todas las pruebas de aceptación. El tester diseñará, aplicará, y documentará una o más pruebas instanciadas (con valores concretos) para cada prueba de aceptación (Letelier Torres, 2010).

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada 'prueba de aceptación'. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.

¹⁹ Investiga un producto de software con el objetivo de obtener información acerca de su calidad y del valor que representa para quienes lo utilizan.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Para eliminar la influencia de conflictos de intereses, y para que sea lo más objetiva posible, la prueba de aceptación nunca debería ser responsabilidad de los ingenieros de software que han desarrollado el producto. Para la preparación, la ejecución y la evaluación de la prueba de aceptación ni siquiera hacen falta conocimientos informáticos. Sin embargo, un conocimiento amplio de métodos y técnicas de prueba y de la gestión de la calidad en general facilita esta labor.

Conclusiones parciales

Para el desarrollo de la herramienta de software que enriquezca una Ontología Geográfica, se representó los componentes mediante un diagrama donde se describe las dependencias existentes entre los nodos físicos en los que funcionará la aplicación, sirviendo como base para la implementación de las funcionalidades. Con el objetivo de tener una aplicación con código legible y entendible para otros usuarios, se hizo uso de los estándares de codificación. Para lograr el funcionamiento de la aplicación, se precisó la realización de las diferentes técnicas de prueba que permiten comprobar la eficiencia del mismo, satisfaciendo así las necesidades del cliente. Con la ejecución de las pruebas de caja blanca y de caja negra, se lograron resultados satisfactorios, evidenciando que la aplicación no presenta ningún error funcional.

CONCLUSIONES GENERALES

Como resultado de la investigación se logró desarrollar una herramienta de software a partir del análisis de bases de datos espaciales y de manera semi-automatizada, permitiendo disminuir el costo en tiempo y esfuerzo del enriquecimiento de ontologías geográficas de gran tamaño; por lo que se llega a las siguientes conclusiones:

- ✓ El análisis de la bibliografía permitió la identificación de las soluciones existentes las cuales sirvieron como base para el desarrollo de la propuesta de solución.
- ✓ Se desarrolló un método para enriquecer ontologías geográficas de manera semiautomático a partir del procesamiento de bases de datos espaciales.
- ✓ La metodología seleccionada para el desarrollo del software es PRODESOF. Se emplea para el modelado UML la herramienta CASE Visual Paradigm, como IDE de desarrollo el NetBeans en su versión 8.0, el lenguaje de programación utilizado fue Java y como marco de trabajo Jena.
- ✓ Para definir las características y condiciones que debe proveer la solución, se identificaron veintitrés requisitos funcionales y dieciocho requisitos no funcionales, estos permitieron una correcta planificación en aras de satisfacer las necesidades del cliente.
- ✓ Con el proceso de construcción del método EDB-to-Geo-Ontology se cumplió con el desarrollo de una herramienta de software capaz de disminuir el costo en tiempo y esfuerzo del enriquecimiento de ontologías geográficas de gran tamaño, a partir del análisis de bases de datos espaciales.
- ✓ Se realizaron las pruebas de caja blanca y negra a la herramienta de software, para probar el correcto funcionamiento de las funcionalidades definidas, considerando la solución obtenida como satisfactoria.

REFERENCIAS BIBLIOGRÁFICAS

Angel Álvarez, Miguel. 2004. desarrollorweb. desarrollorweb. [En línea] 16 de Noviembre de 2004. [Citado el: 16 de Diciembre de 2014.] <http://www.desarrolloweb.com/articulos/1696.php>.

Arano, Silvia. 2003. La ontología: una zona de interacción entre la Lingüística y la Documentación . España : s.n., 2003.

Baglioni, Miriam, y otros. Building Geospatial Ontologies from Geographical Databases .

Barrios, Camelo José Hernández. 2014. slideshare. slideshare. [En línea] 20 de Septiembre de 2014. [Citado el: 25 de Noviembre de 2014.] <http://es.slideshare.net/carmeloh2/metodologa-open-up-39321348>. LinkedIn Corporation.

Bass, L., Clements , P. y Kazman, R. 2003. Software Architecture in Practice. Addison Wesley : 2nd Edition, 2003.

Carabali, Mauricio. 2013. Prezi. Prezi. [En línea] 26 de Septiembre de 2013. [Citado el: 22 de Abril de 2015.] <https://prezi.com/sjwfwmix7slk/pruebas-de-caja-negra-y-caja-blanca/>.

Cervantes, Humberto. 2010. SG. AG. [En línea] Abril de 2010. [Citado el: 10 de Febrero de 2015.] <http://sg.com.mx/revista/27/arquitectura-software>.

Ehrig, Marc. 2007. Ontology Alignment: Bridging the Semantic Gap. Germany : Springer, 2007.

Euzenat, J. y Shvaiko, P. 2007. Ontology Matching. Berlin, New York : Spinger, 2007. 978-3-642-38720-3.

Fensel, Dieter. 2001. Ontologies. Berlin : Springer-Verlag Berlin Heidelberg, 2001. 978-3-662-04398-1.

Fu, G., Jones, C.B, Abdelmoty, A.I. 2005. Building a Geographical Ontology for Intelligent Spatial Search on the Web. En Databases and Applications. 14-16 de Febrero de 2005, p. 167-172.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

García Sánchez, Francisco. 2009. Sistema Basado en Tecnologías del Conocimiento para Entornos de Servicios Web Semánticos. Valencia : s.n., 2009.

Garea Llano, Eduardo , y otros. 2009. SISTEMAS DE INFORMACIÓN GEOGRÁFICA COMO HERRAMIENTA PARA LA INTEGRACION E INTERPRETACIÓN SEMÁNTICA DE LA INFORMACIÓN ESPACIAL. Academia.edu. [En línea] 2009. [Citado el: 14 de diciembre de 2014.] http://www.academia.edu/538903/SISTEMAS_DE_INFORMACION_GEOGRAFICA_COMO_HERRAMIENTA_PARA_LA_INTEGRACION_E_INTERPRETACION_SEMANTICA_DE_LA_INFORMACION.

Garea Llano, Eduardo. 2007. Estado actual de la interpretación semántica de datos espaciales. [Version Digital] s.l. : Serie Azul, 2007. 2142/2072-6287.

Garea Llano, Eduardo, Oliva Santos, Rafael y Costales-Llerandi , Cynthia . 2009. Ontological Integration of Data, Metadata and Knowledge in Geographical Information System for Semantic Interpretation of Spatial Information. 2009.

Gomez Perez, Asuncion, Fernández López, Mariano y Corcho Garcia, Oscar. 2004. Ontological engineering. Computing Reviews. p. 478-479, 2004, Vol. vol. 45, no 8.

González, Yanelis. 2006. Las ontologías en la representación y organización de la información . Las ontologías en la representación y organización de la información . [En línea] 2006. [Citado el: 26 de enero de 2015.] http://bvs.sld.cu/revistas/aci/vol14_4_06/aci08406.htm#cargo.

Gottfried, Konecny. 2014. Geoinformation: remote sensing, photogrammetry and geographic information systems. s.l. : CR Press, 2014.

Gruber, T.R. 1993. A translation approach to portable ontology specifications . Knowledge Acquisition. p. 199-220, 1993, Vol. vol. 5, no 2.

Gruber, Tom. 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. . International Journal of Human-Computer Studies. 1995, Vols. 43(5-6).

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Guarino, N. 1998. Formal Ontology in Information Systems: Proceedings of FOIS '98. Amsterdam : s.n., 1998.

Hendler, James. 2001. Agents and the semantic web. IEEE Computer Society, 2001, Vol. 16, 2.

Hess, Guillermo. 2008. Towards Effective Geographic Ontology. Porto Alegre : s.n., 2008.

Huanca, J. 2015. academia.edu. academia.edu. [En línea] 2015. [Citado el: 3 de Enero de 2015.] http://www.academia.edu/6045417/Visual_Paradigm.

INTECO, Laboratorio Nacional de Calidad del Software de. 2009. INGENIERÍA DEL SOFTWARE:METODOLOGÍAS Y CICLOS. [PDF] España : Laboratorio Nacional de Calidad de Software, 2009.

Larin Fonseca, Rainer y Garea Llano, Eduardo. 2013. Método de Enriquecido Semántico para la Integración de Objetos Geoespaciales. Ciencias de la Tierra y el Espacio, 2013, Vol. 14, No 1.

Larman, Craig. 2004. UML y Patrones. s.l. : Prentice Hall, 2004. ISBN: 8420534382.

Lenguajes de Programacion. 2009. Lenguajes de programación. Lenguajes de programación. [En línea] 2009. [Citado el: 15 de febrero de 2015.] <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.

Letelier Torres, Patricio. 2010. TuneUp Process. TuneUp Process. [En línea] 21 de Octubre de 2010. [Citado el: 10 de Mayo de 2015.] https://www.linkedin.com/grp/post/3636186-48805747?goback=.gde_3636186_member_48805747.

Lopez, F. J., Silva, M. J. y Chaves, M. 2010. Linkable Geographic Ontologies. New York : Proceedings of the 6th Workshop on Geographic Information Retrieval., 2010.

Macías, José A. 2003. Autoría de Documentos Web Dinámicos Mediante Ontologías y Técnicas de Programación por Demostración. 2003.

Maria Jesus. 1999. Herramienta Case. 1999. ISBN: 875-99-OI-OTDETI-INEI.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Martinez Cruz, Carmen, J. Blanco, Ignacio y Amparo Vila, M. 2011. Ontologies versus relational databases: are they so. [PDF] Spain : Springer Science, 2011. Artif Intell Rev (2012) 38:271–290.

Microsoft. 2015. Microsoft. Microsoft. [En línea] 2015. [Citado el: 29 de Marzo de 2015.] <https://msdn.microsoft.com/en-us/library/jj161047.aspx..>

Microsoft. 2014. Microsoft Corporation. Microsoft Corporation. [En línea] 2014. [Citado el: 20 de Marzo de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx..>

Oliva Santos, Rafael y Garea Llano, Eduardo. 2007. Geo-ontologies as Integration Structures of Knowledge, Data, and Metadata. [PDF] Mexico : Second International Conference on Geospatial Semantics., 2007. 78-970-0461-6 (30-11-07).

Oliva Santos, Rafael, Garea Llano, Eduardo y Maciá Perez, Francisco. 2009. Propuesta preliminar de soporte para la integración de datos, metadatos y conocimiento geográfico mediante geo-ontologías. [PDF] España : s.n., 2009. 978-84-613-4894-7.

Peinado Gil, Federico. 2008. Un Armazón para el Desarrollo de Aplicaciones de Narración Automática Basado en Componentes Ontológicos Reutilizables. Madrid : s.n., 2008. ISBN: 978-84-692-0075-9.

Pérez Valdéz, Damián. 2007. Maestros del Web. Maestros del Web. [En línea] 3 de Julio de 2007. [Citado el: 25 de Diciembre de 2014.] <http://www.maestrosdelweb.com/que-es-javascript/>.

PHP. 2014. PHP. PHP. [En línea] 2014. [Citado el: 15 de Diciembre de 2014.] <http://php.net/manual/es/intro-what-is.php>.

Ping, D. y Yong, L. 2009. Building Place Name Ontology to Assist in Geographic Information Retrieval. Computer Science-Technology and Applications, 2009, Vol. International Forum on Volume:1 .

Pressman, Roger S. 5ta Edición. Ingeniería de Software. Un enfoque práctico. s.l. : Mc Graw Hill, 5ta Edición. 5ta Edición.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

- Ramos, E. y Nuñez, H. 2007.** ONTOLOGÍAS: componentes, metodologías, lenguajes, herramientas y aplicaciones. Universidad Central de Venezuela : Lecturas en Ciencias de la Computación, 2007. 1316-6239..
- Rumbaugh, Jacobson. 2000.** El Lenguaje Unificado de Modelado. Manual de referencia. Madrid : Pearson Educación, 2000.
- Sarabia, Gerardo. 2008.** levashkin. levashkin. [En línea] Diciembre de 2008. [Citado el: 15 de Enero de 2015.] <http://levashkin.com/files/Gerardo%20Sarabia%20Lopez.pdf>.
- Smart, P., y otros. 2007.** Toponym ontology Design. 2007.
- Sommerville, Ivan. 2004.** Ingeniería de Software. Madrid : PEARSON EDUCATION S.A, 2004. 84-7820-074-5.
- Tenorio, Robesto Ruiz. 2010.** Cdigital. Cdigital. [En línea] Noviembre de 2010. [Citado el: 29 de Mayo de 2015.] <http://cdigital.uv.mx/bitstream/123456789/28540/1/Ruiz%20Tenorio.pdf>.
- Tolaba, Ana Carolina , Caliusco, Ma. Laura y Galli, Ma. Rosa. 2013.** Meta-ontología Geoespacial: Ontología para Representar la Semántica del Dominio Geoespacial. 2013.
- Tomai, E. y Espanaki, M. 2005.** From ontology design to ontology implementation: A web tool for building geographic ontologies. En Proceedings of AGILE. 2005, Vols. pp. 281-290.
- Torres Rodríguez, Nuria. 2008.** Sistema de análisis automático de fotografías : modelo conceptual según los estándares de la Web Semántica. Getafe : s.n., 2008.
- Torres, Miguel Jesús. 2007.** Representación Ontológica Basada en Descriptores Semánticos Aplicada a Objetos Geográficos. Mexico : Tesis Doctoral. Centro de Investigación en Computación, 2007.
- Vega Ramírez, Alfredo, y otros.** Procedimiento para la obtención de un modelo ontológico para representar la información contenida en bases de datos.

Enriquecimiento de una Ontología Geográfica, por medios semiautomáticos, a partir del Procesamiento de Bases de Datos Espaciales.

Venete, Adrian. 2011. Ingeniería del Software. Ingeniería del Software. [En línea] 18 de Mayo de 2011. [Citado el: 25 de Noviembre de 2014.] <http://mahara.uji.es/view/artefact.php?artefact=54800&view=4648..>

Vera, Francisco y Garea, Eduardo. 2009. Alineamiento de ontologías en el dominio geoespacial. La Habana, Cuba : s.n., 2009. 2072-6287.

WordReference. 2015. WordReference. WordReference. [En línea] 2015. [Citado el: 15 de Abril de 2015.] <http://www.wordreference.com/definicion/componente>.

Zapata, Javier. 2013. Pruebas del Software. Pruebas del Software. [En línea] 21 de Enero de 2013. [Citado el: 12 de Mayo de 2015.] <https://pruebasdelsoftware.wordpress.com>.

Zayas, Carlos Alvarez. 1995. METODOLOGIA DE LA INVESTIGACION CIENTIFICA. Santiago de Cuba : s.n., 1995.