

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



TÍTULO: Módulo para la gestión de metadatos geográficos de LiberMaps

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN INFORMÁTICA**

AUTORES: Aliana López Costa

Liset Garcia Padrón

TUTOR: Ing. Annabell Schelton Lima

CO-TUTORES: MSc. Yusnier Valle Martínez

Ing. Dania Mora Valdés

Ciudad de La Habana

2010

“No basta dar pasos que un día puedan conducir hasta la meta, sino que cada paso ha de ser una meta, sin dejar de ser un paso”

Johann Peter Eckermann (1792-1854)

Dedicatoria

De Aliana

Dedico este trabajo de diploma a mi hermanita Anita, que ya tiene 18 años pero para mí siempre va ser mi hermanita querida y quiero que vea en este triunfo el ejemplo de dedicación y abnegación que siempre he querido ser para ella.

De Liset

Dedico este trabajo con mucho cariño a las personas más importantes de mi vida: mis padres, mi hermano, mi abuelita y mi novio, quienes siempre han estado a mi lado, brindándome todo el apoyo que he necesitado. Con ellos comparto este triunfo, ya que a ellos se lo debo.

Agradecimientos

De Aliana

Quiero agradecer a todos los que han ayudado a mi formación como profesional, en especial:

A mis padres Regla y José Luis por ser mi lucero guía desde el día que nací, por todo el amor que me han dado, por ser mi ejemplo, por darme todo su apoyo incondicional, por sentir mis logros, derrotas, los momentos tristes, alegres y de tensión como si fueran de ellos también, en fin por haber sido para mí el sustento fundamental de mi vida.

A mi hermanita Anita por ser mi apoyo, mi confidente, por todo el amor que siento por ella y por el que siente ella por mí, por ser esa personita por la que siempre me he esforzado para que viera en mí un ejemplo a seguir.

A mi abuela Alicia por adorarme desde el día que llegue a este mundo y porque yo también la quiero a ella con el alma, por estar pendiente de mis triunfos y derrotas y hacérmelo saber, por estar siempre en mis pensamientos y aun en la distancia hacerse sentir cerca de mí en todo momento.

A mis demás familiares también por su preocupación por mí y por estar pendientes de mi carrera universitaria.

A mi novio Diuber, que más que mi pareja ha sabido ser mi amigo, me ha brindado un inmenso amor, apoyo y dedicación, por aguantarme mis malcriadeces durante estos cinco años, por ser el hombro que he tenido para llorar, en fin por ser el amor de mi vida.

A Liset que ha sido mi amiga durante estos cinco años y tuve la suerte de tenerla como compañera de tesis, por aguantar mis arrebatos, mis malcriadeces, por hacer posible este sueño junto a mí.

A mis amigas del pre y de siempre Madelayne, Katia y Lianet por estar pendientes de mí aunque lejos y por confiar en mí y quererme tanto.

A mis grandes amigos de la misión que cumplí en el hermano país de Venezuela María Ester y Juan Carlos por haber sido mi familia en esos momentos tan duros para mí, por preocuparse y confiar en mí aun en la distancia.

A mis vecinos del edificio Rubí, Reinier, Edita, Pedro y todos los demás, por estar también siempre pendientes de mí y de mis estudios.

De Liset

A mi mamá Cari, por ser mi confidente y amiga, por tener el valor de rectificarme cuando me equivoco, por aconsejarme cuando tengo dudas y apoyarme siempre que la he necesitado. Dicen que todas las madres son buenas, pero para mí tú eres la mejor, te quiero...

A mi papá Jose Luis, por estar siempre a mi lado y preocuparse de que nunca me faltara nada, por darme la confianza que he necesitado de su parte para poder seguir adelante, a quien respeto, admiro y quiero muchísimo, y he aprendido a verlo, más que como a un padre, como a un amigo.

A mi hermano Robertico, que aunque es mayor que yo, le tengo el cariño que se le tiene a un hermano menor, ha sido mi apoyo en muchos momentos, mi consejero, pero también ha sabido darme un ejemplo de la profesional que debo llegar a ser.

A mi abuelita Valeria, por darme tanto amor y a quien quiero darle la alegría de verme graduada, es la mujer más fuerte y cariñosa que he conocido y quisiera llegar a ser así algún día.

A mi novio, mi confidente y mejor amigo durante más de 5 años, mi soporte en varios momentos difíciles de mi vida, me ha apoyado siempre demostrándome que soy capaz de superar cualquier obstáculo, ha aguantado mis malcriadeces y mis regaños más inesperados, pero siempre con el buen humor que lo caracteriza...te amo tatiko.

A mis primos y tíos, por estar siempre al tanto de lo que sucede en mi vida y darme ánimos para seguir adelante.

A mi amiga y compañera de tesis Aliana, quien me animó y me dio confianza en todo momento, ha sido un gran apoyo durante los 5 años de la carrera y gracias a su empeño y dedicación hoy disfrutamos juntas el fruto de nuestro trabajo.

A todas las amistades que me he ganado en esta escuela, muchas de las cuales tal vez no vuelva a ver por la distancia que nos separa, pero que siempre las tendré presentes en mi corazón

A mis grandes amigas y en algún momento compañeras de cuarto Aliana, Yanita y Lisandra, quienes me hicieron sentir en su compañía como con mi propia familia, depositaron en mí su confianza y supieron ganarse la mía con facilidad.

A todas estas personas especiales para mí, quiero darle las gracias por brindarme toda su ayuda, apoyo y comprensión, alentándome de esta forma a para lograr hacer mi sueño realidad.

De ambas

A nuestra tutora Annabell, por ser nuestra amiga, por saber guiarnos por el camino correcto y sufrir junto a nosotras todos los momentos tensos y por estar siempre pendientes de nosotras aun estando lejos.

A nuestros cotutores Valle y Dania por toda la ayuda que nos brindaron con la mayor disposición posible.

A los muchachos de UCIFAR Rolando, Membrides, Manuel, Karel, y a todos los demás sin excepción por su ayuda incondicional siempre con nosotras y por hacernos reír hasta en los momentos más tensos.

A los muchachitos de segundo año Yadir, Jorge y Javier por toda la ayuda que nos brindaron y su preocupación en todo momento por nuestro trabajo.

A nuestro Tribunal por todo el apoyo que nos brindaron, por sus críticas constructivas que hicieron posible la culminación de este trabajo con la mayor calidad posible.

A todos nuestros compañeros de aula, los del grupo de primer año y los de ahora y a todas las muchachitas que han estado con nosotras en los apartamentos desde primer año, por haber sido nuestra familia durante estos 5 años de nuestra vida.

A todos los profesores que nos han impartido clases en estos 5 años por haber contribuido a nuestra formación como profesionales.

Finalmente a todos aquellos que nos han ayudado a hacer realidad este sueño tan anhelado por ambas.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Liset Garcia Padrón

Ing. Annabell Schelton Lima

Aliana López Costa

DATOS DE CONTACTO

TUTOR: Ingeniera Annabell Schelton Lima (email: aschelton@uci.cu). Profesora graduada de Ingeniería en Ciencias Informáticas.

CO-TUTOR: Yusnier Valle Martínez (email: yvm@uci.cu). Máster en Ciencias graduado de Licenciatura en Ciencias de la Computación.

CO-TUTOR: Ingeniera Dania Mora Valdés (email: dmora@uci.cu). Profesora graduada de Ingeniería en Ciencias Informáticas.

OPINIONES Y AVALES



CARTA DE AVAL SOBRE RESULTADO DE TRABAJO

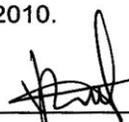
A quien pueda interesar:

El Trabajo de Tesis de Grado titulado "Módulo para la gestión de metadatos geográficos de LiberMaps", fue realizado en la Universidad de las Ciencias Informáticas en el período de octubre de 2009 a junio de 2010 por las autoras Aliana López Acosta y Liset García Padrón. Dicho trabajo fue desarrollado en la línea de desarrollo del proyecto GeneSIG, constituyendo un complemento más de los módulos especializados del paquete tecnológico de la herramienta de igual nombre y aportando los siguientes resultados:

- Permite documentar la información geográfica y facilita una mejor organización y localización de los datos de cualquier infraestructura o entidad de la rama Geomática.
- Posibilita explotar los beneficios de la red mediante su arquitectura de sistema Web, minimizando los costos de hardware en las organizaciones cubanas y centralizando la información en unidades de datos seguras.
- Complementa, como solución de software, el sistema de catálogo Libermaps, brindando a los usuarios una herramienta importante para la gestión de la información atributiva de la confección, actualización y soporte de la información geográfica.

El aporte de este desarrollo, además, le permite al país contar con una herramienta propia que responda a los intereses de sus organizaciones, y que esté respaldada por un equipo de desarrollo cubano que pueda dar soporte y mantenimiento a esta solución, problema muy común en la comunidad de software libre.

Y para que así conste, se firma la presente a los 21 días del mes de junio del año 2010.




Ing. Yoenis Pantoja Zaldívar

Jefe de Departamento Geoinformática. Facultad 9. UCI

Facultad 9



“El futuro de nuestra Patria tiene que ser, necesariamente, un futuro de hombres de ciencia, de hombres de pensamiento.”

Fidel Castro Ruz, 1960.

XVI FÓRUM DE CIENCIA Y TÉCNICA

Se otorga la categoría de **RELEVANTE**

Al trabajo:

LibertHaps: Sistema de catálogo para la gestión y metadatos en la web

De los autores:

Yosnier Valle Martínez, Dania Mora Valdés
Annabell Schelton Lima, Yadir Hernández
Aliana López Costa, Jorge Fuentes, Javier Rojas
Liset García Padrón, Grethell Castillo

RESUMEN

En la Universidad de las Ciencias Informáticas (UCI) se desarrollan proyectos productivos que permiten a los estudiantes vincular lo aprendido en clases con la producción y al mismo tiempo ayudar a informatizar varios sectores de la sociedad y a contribuir con países hermanos. Como parte de la realización de estos proyectos un grupo de trabajo de la facultad 9 de la UCI está desarrollando un SIG que lleva por nombre GeneSIG. El mismo tiene integrado el catálogo de mapas LiberMaps el cual maneja la seguridad sobre datos y funcionalidades, permite realizar la edición dinámica de los objetos de un mapa y poseer una estructura organizativa para estos, pero no cuenta con una documentación organizada de los datos contenidos en ellos, problema cuya solución la constituyen los metadatos geográficos que no es más que una descripción de datos geoespaciales específicos, que permite distinguirlos a través de aspectos como son: calidad, actualización, autor, entre otros.

Es por ello que surge la necesidad de implementar un sistema informático que permita gestionar los metadatos geográficos de LiberMaps, proporcionando la información necesaria para interpretar y procesar los datos de este. Para ello se llevó a cabo un estudio de los procesos de gestión de metadatos geográficos y de las tendencias actuales en el mundo de la informática. El sistema está desarrollado en un ambiente web con tecnologías libres en su mayoría. Teniendo en cuenta la metodología seleccionada: RUP, se documentó todo el proceso generando los artefactos correspondientes a cada flujo de trabajo.

PALABRAS CLAVES

Ambiente web, Datos geoespaciales, Gestión, Mapa, Metadatos, Sistema informático, Tecnologías libres.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación Teórica	6
1.1. Introducción.....	6
1.2. Conceptos asociados al dominio del problema.....	6
1.2.1. Información geográfica	6
1.2.2. Dato espacial.....	6
1.2.3. Sistema de Información Geográfica	7
1.2.4. Infraestructura de Datos Espaciales	7
1.2.5. Normas.....	7
1.2.6. ISO	7
1.2.7. Catálogo de mapas.....	7
1.2.8. Mapa	8
1.2.9. Metadato geográfico	8
1.3. Estado del arte	8
1.3.1. Normas internacionales	8
1.3.1.1. Normas ISO para metadatos geográficos	9
1.3.1.1.1. ISO 19115: Metadatos geográficos	9
1.3.1.1.2. ISO 19139: Metadatos. Esquema de implementación XML.....	11
1.3.1.1.3. ISO 19115-2: Metadatos para imágenes y datos en malla	11
1.3.1.1.4. Perfiles de aplicación de la norma ISO 19115.....	12
1.3.1.1.4.1. ISO 19115 NEM (Núcleo Español de Metadatos).....	12
1.3.1.1.4.2. ISO 19115 INSPIRE	13
1.3.1.1.4.3. ISO 19115 WISE	14
1.3.2. Ámbito de Cuba	15
1.3.3. Definiendo la norma a utilizar.....	16
1.4. Situación Problemática.....	17
1.5. Análisis de otras soluciones existentes.....	17
1.6. Conclusiones.....	19
CAPITULO 2. Tendencias y tecnologías actuales a utilizar	20
2.1 Introducción.	20
2.2 Metodologías de Desarrollo de Software.....	20
2.2.1 Rational Unified Process (RUP).....	21
2.3 Lenguaje Unificado de Modelado (UML).....	22
2.4 Servidores Web.....	22
2.4.1 Servidor HTTP Apache.....	22
2.5 Lenguajes de Programación Web.....	23

2.5.1	Lenguajes del lado del servidor.	23
2.5.2	Programación del lado del cliente.....	24
2.6	Herramientas.	25
2.6.1	Sistemas Gestores de Bases de Datos.	26
2.6.1.2	PostgreSQL.....	26
2.6.2	Herramientas CASE.	28
2.6.2.2	Visual Paradigm.	28
2.6.3	Framework.	29
2.6.3.2	Framework Symfony.....	29
2.6.4	Entorno Integrado de Desarrollo (IDE).	30
2.6.4.2	NetBeans.....	31
2.6	Justificación de la selección.	31
2.7	Conclusiones.....	33
CAPÍTULO 3: Presentación de la solución propuesta.....		34
3.1.	Introducción	34
3.2	Modelo de dominio.....	34
3.2.1.	Diagrama de clases del Modelo de Dominio.....	34
3.2.2.	Glosario de Términos del Dominio.....	35
3.3.	Modelo del Sistema.....	36
3.3.1.	Requisitos Funcionales	36
3.3.2.	Requisitos no Funcionales.....	39
3.3.3.	Diagrama de Casos de Uso del Sistema (DCUS).	42
3.3.4.	Descripción de los actores del sistema.	43
3.3.5.	Descripción de los Casos de uso del Sistema.	44
3.4	Conclusiones	47
CAPÍTULO 4: Construcción de la solución propuesta.		48
4.1.	Introducción	48
4.2.	Diseño.....	48
4.2.1.	Descripción general del funcionamiento del Symfony y ExtJs.	48
4.3.	Patrones de Diseño.....	51
4.3.1.	Patrones GRASP implementados en Symfony.....	51
4.3.2.	Tipos de patrones GOF que implementa Symfony	52
4.3.3.	Patrones implementados en la vista de la aplicación.....	52
4.4.	Diagramas de Clases del Diseño.	53
4.4.1.	Realización del Caso de Uso: Gestionar Metadato.....	53
4.4.1.1.	Paquete Modelo del Caso de Uso Gestionar Metadato.	54
4.5.	Diseño de la Base de Datos.	54
4.5.1.	Diagrama de Clases Persistentes.....	54

4.5.2. Modelo Entidad-Relación	56
4.6. Modelo de Despliegue.....	57
4.7. Modelo de Implementación	57
4.8. Conclusiones	60
CONCLUSIONES	61
RECOMENDACIONES.....	62
REFERENCIAS BIBLIOGRÁFICAS.....	63
BIBLIOGRAFÍA.....	68
GLOSARIO DE TÉRMINOS	70

INTRODUCCIÓN

Los mapas desde hace siglos han sido usados por las personas para representar el medio donde viven. Un mapa es un dibujo sobre un plano que representa un área geográfica. En la actualidad, en casi todas las esferas de la ciencia se utilizan y es por ello que se han convertido en un instrumento imprescindible en muchos aspectos profesionales y académicos.

Con la revolución de la información llegaron las primeras aplicaciones cartográficas realizadas en computadoras del decenio de 1960. Estas no tuvieron mucha aplicación excepto en unos pocos proyectos oficiales y académicos, pues era de gran complejidad representar digitalmente los mapas en aquellas máquinas, ya que requerían de una gran capacidad para almacenar datos y rápidos recursos informáticos. Ya en el decenio de 1980 los sistemas comerciales de información geográfica lograron la capacidad que permitiría su rápida aceptación.

Los avances tecnológicos de finales del siglo pasado dieron lugar al surgimiento de los Sistemas de Información Geográfica (SIG) y estos a su vez influyeron potencialmente en que en la actualidad haya una mayor accesibilidad de la información geográfica. Un SIG es un sistema computarizado que permite a los usuarios gestionar y analizar información geográfica. El uso de los SIG ha permitido a personas y organizaciones analizar hechos y oportunidades y a su vez resolver problemas, por lo que representan una herramienta fundamental de análisis y de toma de decisiones.

Los SIG permiten asociar cualquier tipo de información con una ubicación geográfica. Gracias a estos, médicos y hospitales pueden relacionar mapas de enfermedades con condiciones de salud, las autoridades pueden relacionar mapas de lugares donde se cometieron crímenes con patrones de criminalidad, el personal de servicios de emergencia puede relacionar mapas de áreas de riesgo con información sobre inundaciones o incendios forestales.

Los adelantos en distintas áreas de la informática han contribuido significativamente al avance de los SIG, por ejemplo, las bases de datos facilitan la gestión de grandes cantidades de información que sirven de referencia para los mapas digitales.

El avance en el desarrollo de los SIG junto a los logros en otros campos de la ciencia, han hecho que el número, complejidad, calidad y diversidad de los datos geográficos crezca aceleradamente, por lo que fue

necesario crear métodos que permitieran agrupar todos estos elementos de forma tal que se lograra un manejo eficiente de los datos.

Se hacía necesario contar con una documentación organizada de la información existente y de la que se creara posteriormente con el objetivo de facilitar las búsquedas de esta información, poder caracterizarla y analizarla sin mucha dificultad. La solución a este problema la constituyeron los metadatos. Los metadatos geográficos no son más que una descripción de datos geoespaciales específicos que permite distinguirlos a través de aspectos como son: calidad, actualización, referencia geoespacial, autor, entre otros.

Cuba también se está adentrando en el mundo tecnológico actual, como ejemplo de ello está la creación de la Universidad de las Ciencias Informáticas (UCI). En esta se desarrollan proyectos productivos que permiten a los estudiantes vincular lo aprendido en clases con la producción y al mismo tiempo ayudar a informatizar varios sectores de la sociedad y a contribuir con países hermanos.

Como parte de la realización de estos proyectos un grupo de trabajo de la facultad 9 de la UCI está desarrollando un SIG que lleva por nombre GeneSIG. Este es un producto soberano que sirve como apoyo al desarrollo de aplicaciones de SIG en entornos Web con tecnologías libres.

Uno de los principales componentes que son necesarios para la construcción de una Infraestructura de Datos Espaciales (IDE) son los catálogos de información geográfica. Las IDEs pretenden catalogar y poner al alcance de todos, datos georreferenciados que, con frecuencia, son desconocidos o no son dados a conocer de una forma adecuada.

Los catálogos permiten que usuarios o aplicaciones software puedan buscar la información existente en algún lugar dentro de un entorno computacional distribuido. Estos son la solución para publicar descripciones de datos geográficos mediante métodos estándar que posibilitan la realización de búsquedas a lo largo de múltiples servidores y además son una herramienta útil para integrar con los Sistemas de Información Geográfica.

La plataforma GeneSIG como SIG necesitaba contar con una lista ordenada o clasificada de los mapas contenidos en ella y por ello había que adicionarle un módulo que facilitara esto. Así surgió el catálogo de mapas LiberMaps que además posee vida propia como software independiente.

Este catálogo maneja la seguridad sobre datos y funcionalidades, permite realizar la edición dinámica de los objetos de un mapa y poseer una estructura organizativa para estos, de modo que el usuario pueda

interactuar con ellos más organizadamente. En el mismo, como en todo catálogo de mapas, se trabaja con grandes cantidades de información geográfica por lo que se observó la necesidad de documentar toda esta información y facilitar una mejor organización y localización de los datos. Era preciso mantener el acervo del conjunto de datos contenidos en el mismo y proporcionar la información necesaria para interpretar y procesar los datos de este.

Por todo lo anteriormente expuesto se decidió realizar esta investigación identificando como **problema a resolver** la Necesidad de clasificar, ordenar y facilitar el acceso a los mapas contenidos en el catálogo de mapas LiberMaps.

A partir del problema anterior se define entonces como **objeto de estudio** los Procesos de gestión de metadatos geográficos.

Se determina como **objetivo general** de la investigación Desarrollar un módulo para el catálogo de mapas LiberMaps que permita la gestión de metadatos geográficos.

El **campo de acción** lo constituye la Automatización de los procesos de gestión de metadatos geográficos.

Teniendo en cuenta todo lo anterior se plantea como **idea a defender** Contar con un sistema que gestione los metadatos asociados a los mapas contenidos en el catálogo LiberMaps, permitirá acceder a los mismos de manera más eficiente y rápida.

Para dar cumplimiento al objetivo de esta investigación se definieron las siguientes **tareas investigativas**:

- Identificar las normas para la creación de metadatos geográficos.
- Definir la norma a utilizar para la creación de los metadatos geográficos.
- Identificar las tendencias en el desarrollo de sistemas para la gestión de metadatos.
- Determinar las tecnologías sobre las que se desarrollará el módulo para la gestión de metadatos.
- Identificar requisitos funcionales y no funcionales para desarrollar el módulo para la gestión de metadatos.
- Diseñar el módulo para la gestión de metadatos.
- Implementar el módulo para la gestión de metadatos.

- Documentar la investigación con vistas a su socialización.

Para desarrollar estas tareas se utilizarán los siguientes **métodos científicos**:

Métodos Teóricos:

Análisis histórico - lógico para el análisis histórico de los procesos de gestión de metadatos geográficos y de las tendencias actuales en el desarrollo de sistemas de este tipo.

Análisis y síntesis para realizar una revisión bibliográfica de todos los temas asociados a la investigación y luego hacer una recopilación de la bibliografía a utilizar.

Modelación para realizar diagramas y representar todo el proceso de la gestión de los metadatos geográficos en el catálogo LiberMaps facilitando un mejor entendimiento de la solución a implementar.

Métodos Empíricos:

Observación utilizada durante todo el proceso de investigación, para identificar todo lo relacionado con el dominio del problema.

Se espera como **posible resultado** de esta investigación contar con una Herramienta para gestionar los metadatos geográficos asociados a los mapas contenidos en el catálogo LiberMaps.

La investigación estará estructurada de la siguiente manera:

Capítulo 1. Fundamentación Teórica: En el mismo se hace referencia a conceptos asociados al dominio del problema así como un estudio del estado del arte y las soluciones existentes en el mundo.

Capítulo 2. Tendencias y tecnologías actuales a utilizar: Se realiza un análisis de las tecnologías y metodologías para la construcción de software existentes en la actualidad y se reflejan las que se van a utilizar en el sistema a implementar.

Capítulo 3. Presentación de la solución propuesta: Se analizan los procesos de gestión de metadatos geográficos para describir la propuesta del sistema. Se determinan los conceptos principales para realizar

el modelo de dominio y se identifican los requisitos funcionales y no funcionales del sistema representándose en el diagrama de casos de uso del sistema.

Capítulo 4. Construcción de la solución propuesta: Se modela el diseño del sistema y se plantea el modelo de datos. Se efectúa la implementación del sistema, reflejándose los diagramas correspondientes como son el de componentes y el de despliegue.

CAPÍTULO 1: Fundamentación Teórica.

1.1. Introducción

En este capítulo se abordan distintos temas que permiten justificar teóricamente la investigación, entre ellos los conceptos asociados al dominio del problema como qué son los metadatos y específicamente los metadatos geográficos, qué es un mapa, cómo se definen las normas, y otros. Se analizan además, las normas internacionales que existen para la creación de metadatos geográficos, así como lo referente a estas en Cuba. Se brinda una explicación de la problemática que condujo a esta investigación y se hace un análisis de los sistemas gestores de metadatos existentes en el mundo arribando a conclusiones sobre cuál guía el trabajo realizado y la norma utilizada.

1.2. Conceptos asociados al dominio del problema

De forma tal que se puedan comprender mejor los principales aspectos que conforman esta investigación es importante conocer algunos conceptos asociados al dominio del problema.

1.2.1. Información geográfica

La **información** es un conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo. **(Thompson, 2008)**

Se puede concluir entonces que la **información geográfica** es un conjunto de datos espaciales que brindan conocimiento sobre algún suceso, hecho o fenómeno.

1.2.2. Dato espacial

Un **dato espacial** es un dato que además de tener una serie de características como color, forma y otras, posee también una ubicación en el espacio (coordenadas x, y, z).

1.2.3. Sistema de Información Geográfica

El profesor David Rhind (1989) plantea que un **Sistema de Información Geográfica (SIG)** es un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados (georreferenciados), para la solución de los problemas complejos del manejo y planeamiento territorial. **(Cisneros Díaz, y otros, Julio de 1994)**

1.2.4. Infraestructura de Datos Espaciales

La **Infraestructura de Datos Espaciales -IDE** se concibe como el conjunto de organizaciones que definen e implementan una serie de estrategias que permiten optimizar la producción y gestión de información geográfica, a fin de facilitar tanto su acceso como su uso para la toma de decisiones. **(Bohorquez, 1999)**

1.2.5. Normas

Según la Real Academia Española las **normas** se definen como “reglas que se deben seguir, o a las que se deben ajustar las conductas, tareas, actividades.” **(Real Academia Española)**

1.2.6. ISO

ISO (La Organización Internacional de Normalización) es una federación mundial de organismos nacionales de normalización (organismos miembros de ISO). El trabajo de preparación de las normas internacionales se suele realizar a través de los comités técnicos de ISO. **(Secretaría del CTN50. AENOR. FESABID, 2008)**

Las Normas Internacionales se proyectan de acuerdo con las reglas establecidas en la Parte 2 de las Directivas ISO/CEI. **(Secretaría del CTN50. AENOR. FESABID, 2008)**

1.2.7. Catálogo de mapas

Un **catálogo** es una relación ordenada en la que se incluyen o describen de forma individual libros, documentos, personas, objetos y otros elementos, que están relacionados entre sí. **(Real Academia Española)**

Entonces se puede decir que un **catálogo de mapas** es una relación ordenada de mapas que permite facilitar su localización en caso de ser necesitada.

1.2.8. Mapa

Un **mapa** es una representación geográfica de una parte de la superficie terrestre, en la que se presenta información relativa a una ciencia determinada. **(Real Academia Española)**

1.2.9. Metadato geográfico

Un **metadato** es una información que describe entre otras, la calidad, distribución, actualidad y referencia espacial de un conjunto de datos. Muchos especialistas llaman a los metadatos como “datos acerca de los datos”. **(COMISIÓN DE LAS COMUNIDADES EUROPEAS, 2008)**

Un **metadato geográfico** trata de caracterizar los datos espaciales respondiendo al quién, qué, cuándo, dónde, por qué y cómo de estos datos.

Los metadatos para que sean realmente importantes deben estar compilados en bases de datos y disponibles en Servicios de Consulta de Información Geográfica o ClearingHouse¹.

1.3. Estado del arte

1.3.1. Normas internacionales

Internacionalmente se han definido normas o estándares para la creación de metadatos, cuyo objetivo principal es brindar una estructura para describir o caracterizar cada uno de los datos geográficos que se

¹ Un ClearingHouse es una red distribuida de productores y usuarios de datos espaciales que permite encontrar y acceder a metadatos y datos geográficos, con escasas restricciones.

referencian. Estas normas han sido aprobadas por expertos en esta materia que pertenecen a organizaciones de normalización.

La ISO a través de su familia ISO 19100 define normas relacionadas con metadatos.

1.3.1.1. Normas ISO para metadatos geográficos

Actualmente el uso de la información geográfica se ha incrementado, cada vez acceden a la misma mayor cantidad de usuarios y con diferentes propósitos, la mayoría de las veces esta información no viene acompañada de datos que la representen como su autor, modo de identificación, fecha, entre otras características. Es por ello que se hace necesaria una estructura bien organizada que permita su documentación, la cual está basada en la utilización de los metadatos. Existen estándares internacionales que definen los elementos fundamentales que deben contener los metadatos geográficos y establecen un grupo de términos comunes.

1.3.1.1.1. ISO 19115: Metadatos geográficos

Esta norma brinda una guía y define un conjunto común de términos, conceptos y procedimientos que se aplican a los metadatos. Los elementos de metadatos describen la información sobre la identificación, la extensión, la calidad, el modelo espacial y temporal, la referencia espacial y la distribución de los datos geográficos.

Esta norma tiene aplicaciones como:

- La catalogación de conjuntos de datos, actividades de clearinghouse, y la descripción completa de conjuntos de datos.
- Diferentes niveles de información: conjuntos de datos geográficos, series de conjunto de datos, fenómenos geográficos individuales, propiedades de los fenómenos y otros. **(9)**

Este estándar de metadatos es muy complejo, cuenta con una gran cantidad de elementos de metadatos, unos obligatorios y otros opcionales. En él se definen 27 listas controladas, a partir de las cuales se especifican los posibles valores que pueden tomar determinados campos.

Los metadatos geográficos se muestran mediante paquetes UML como aparece en la figura 1. Dentro de cada paquete hay una o más entidades (clases UML), que pueden estar especificadas o generalizadas y cada una de ellas se puede relacionar con otras dentro del mismo paquete.

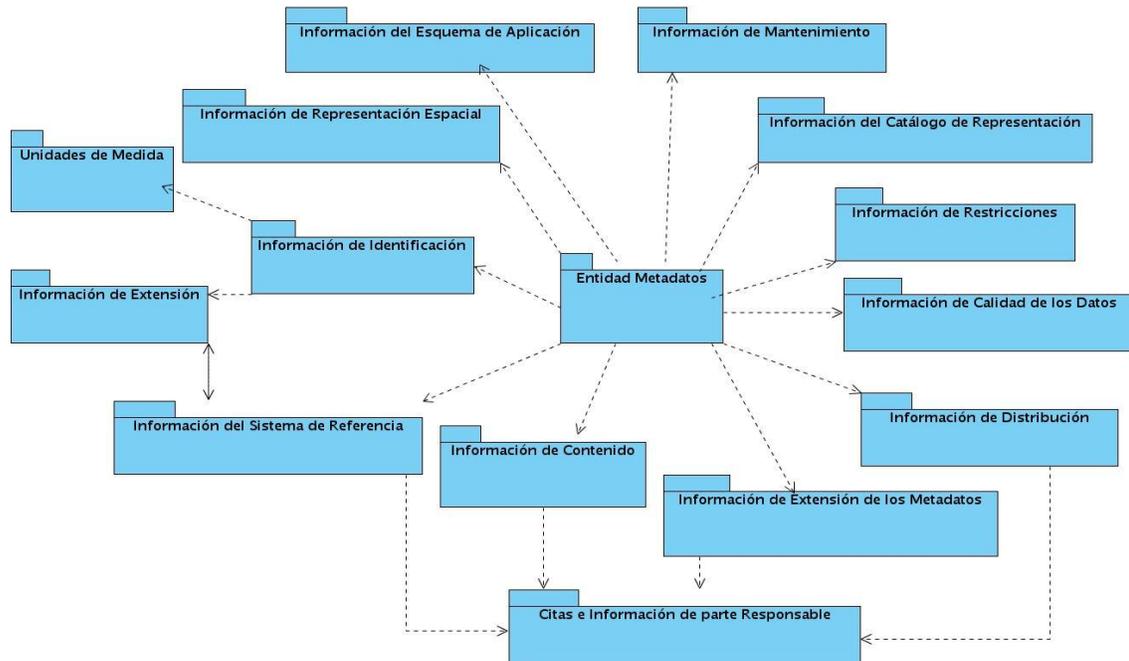


Figura 1: Relaciones entre paquetes UML.

Esta norma, a pesar de ser muy extensa, cuenta con un “conjunto mínimo” de metadatos que se denomina núcleo o Core, cuyos elementos se muestran en la figura 2, que es de obligatoria implementación para todas las aplicaciones de metadatos (desde mapas en formato papel a datos en formato digital, como imágenes satélite, modelos digitales del terreno, entre otros), y al utilizar todos sus elementos (opcionales y obligatorios) se incrementa la interoperabilidad de los datos y ayuda a los usuarios a entenderlos mucho mejor. Con esta mínima selección se pretende proveer mejoras en el descubrimiento, el acceso, la transferencia y la utilización de los datos.

Elementos Obligatorios del Núcleo ISO	Elementos Opcionales del Núcleo de ISO
<ul style="list-style-type: none">• Título del Conjunto de Datos• Fecha de Referencia• Idioma del Conjunto de Datos• Categoría del tema• Resumen descriptivo• Fecha de los Metadatos	<ul style="list-style-type: none">• Parte responsable del Conjunto de Datos• Resolución espacial del conjunto de datos• Formato de Distribución• Extensión vertical y temporal• Tipo de representación espacial• Sistema de Referencia• Linaje• Recurso en línea• Identificador del Fichero de Metadatos• Norma de Metadatos• Versión de la Norma de Metadatos
Elementos Condicionales del Núcleo de ISO	
<ul style="list-style-type: none">• Localización geográfica de los Datos• Conjunto de caracteres del Conjunto de Datos• Idioma de los Metadatos• Conjunto de caracteres de los Metadatos• Punto de contacto para los Metadatos	

Figura 2: Elementos del núcleo de la norma ISO19115

1.3.1.1.2. ISO 19139: Metadatos. Esquema de implementación XML

ISO 19139 es una especificación técnica que desarrolla una implementación en XML del modelo de metadatos descrito por ISO 19115.

Un archivo de intercambio de metadatos, acorde con el estándar ISO19115 y en formato XML, es un documento XML que sigue la sintaxis definida por la especificación técnica ISO19139 (Maganto, 2008). Es decir, en esta especificación técnica se definen un conjunto de esquemas en XML que detallan los metadatos correspondientes a cada nivel de información, de esta forma se logra su descripción, validación e intercambio de archivos.

1.3.1.1.3. ISO 19115-2: Metadatos para imágenes y datos en malla

En los Sistemas de Información Geográfica, una importante fuente de información son las imágenes de satélite, orto-fotografías, modelos digitales y otros. Estos datos tienen características exclusivas, que deben mostrarse mediante los metadatos.

El objetivo de esta norma es definir los metadatos asociados a datos ráster y malla que van a contener información sobre:

- Las propiedades de los equipos de medición usados para adquirir los datos.
- La geometría de los procesos de medición empleados por los equipos.
- Los procesos de producción usados para obtener los datos.
- Los métodos numéricos y procesos informáticos usados. **(Maganto, 2008)**

En esta norma los metadatos son mostrados mediante paquetes UML que incluyen las entidades que contienen los elementos. También incluye un “Diccionario de Datos” que encierra las definiciones de las entidades y elementos para los esquemas de metadatos adicionales definidos.

1.3.1.1.4. Perfiles de aplicación de la norma ISO 19115.

El modelo general propuesto por ISO19115 (409 elementos) resulta complejo para algunas organizaciones a la hora de adoptarlo directamente como guía para la descripción de los recursos de información geográfica.

Debido a esta dificultad, frecuentemente se recurre a perfiles de ISO 19115 que, sin dejar de tener en cuenta las reglas especificadas por esta norma, sintetizan el modelo general de ISO19115. Estos perfiles responden a necesidades de una entidad, organización o infraestructura determinada.

1.3.1.1.4.1. ISO 19115 NEM (Núcleo Español de Metadatos)

Núcleo Español de Metadatos (NEM) es una muestra de la implementación de un perfil de la norma ISO 19115 en España.

El NEM es un perfil de metadatos de ISO 19115: 2003 de acuerdo con el concepto de perfil definido en la Norma ISO 19106. **(Maganto, 2008)**

Este perfil incluye una serie de elementos de la norma ISO 19115 y de otras como se muestra en la figura 3.

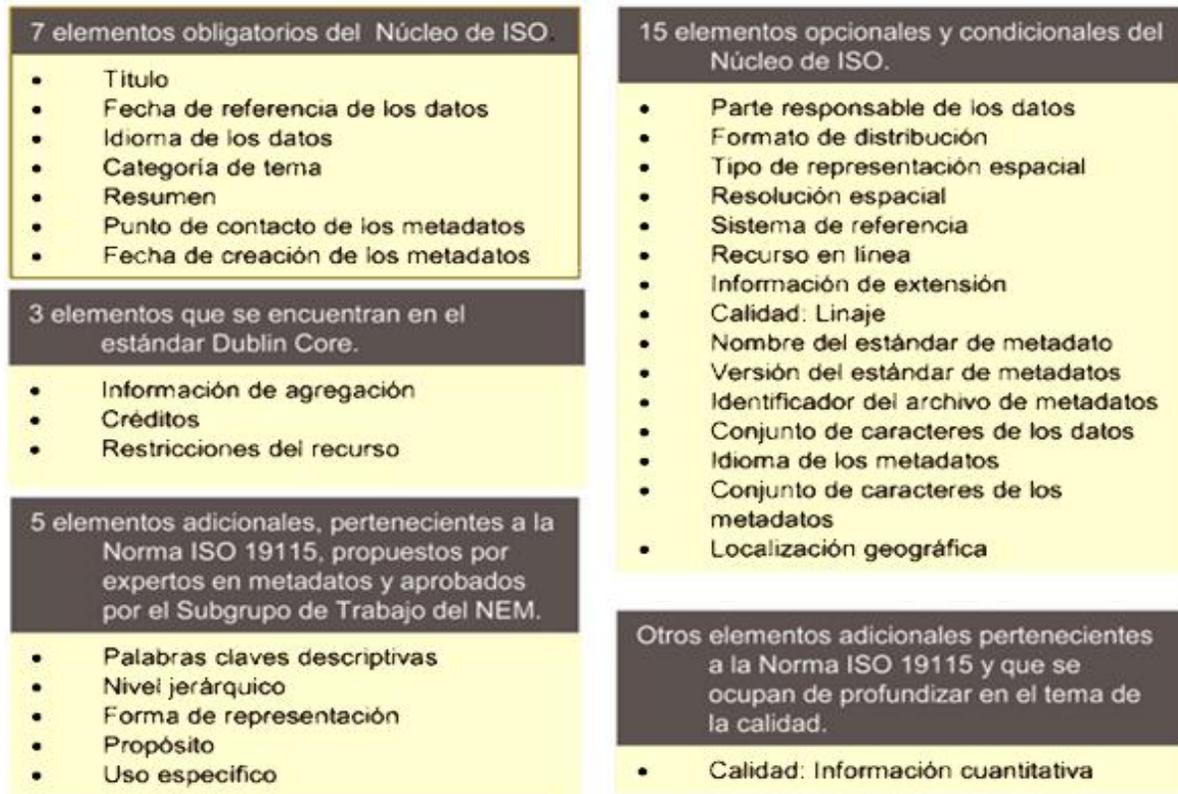


Figura 3: Elementos del NEM.

1.3.1.1.4.2. ISO 19115 INSPIRE

La directiva INSPIRE² brinda un conjunto de elementos de metadatos para el descubrimiento, que se corresponden tanto con la norma ISO 19115 como con ISO-19119 y ISO-15836 (Dublín Core). ISO-19115 aporta la terminología necesaria para describir los datos geográficos a nivel de capa, serie o servicio; mientras que ISO-19119 amplía la descripción de los servicios de información geográfica.

INSPIRE también trabaja en la estandarización de los metadatos geográficos de las Administraciones Públicas de los estados miembros de la Unión Europea, las especificaciones que utilizan para esto están incluidas en sus Reglas de Implementación (IR).

² Infraestructure for Spatial Information in Europe: es una iniciativa de la Comisión Europea cuyo funcionamiento se recoge en la Directiva 2007/2/CE del Parlamento Europeo y del Consejo, de 14 de marzo de 2007, publicada en el Diario Oficial de la UE (DOUE) el 25 de Abril de 2007, que tiene como objetivo la creación de una Infraestructura de Datos Espaciales en Europa.

Este perfil incluye un conjunto de elementos de metadatos geográficos como se muestra en la figura 4.

Identificación <ul style="list-style-type: none">• Título del recurso• Resumen del recurso• Tipo del recurso• Localizador del recurso• Identificador único de recursos• Recurso acoplado• Lengua del recurso	Clasificación de Datos y Servicios Espaciales <ul style="list-style-type: none">• Categoría temática• Tipo de servicio de datos espaciales
Localización Geográfica <ul style="list-style-type: none">• Rectángulo geográfico envolvente	Palabra Clave <ul style="list-style-type: none">• Valor de la palabra clave• Vocabulario controlado de origen
Calidad y Validez <ul style="list-style-type: none">• Linaje• Resolución espacial	Referencia Temporal <ul style="list-style-type: none">• Extensión temporal• Fecha de publicación• Fecha de la última revisión• Fecha de creación
Constricciones Relacionadas con el Acceso y el Uso <ul style="list-style-type: none">• Condiciones aplicables al acceso y el uso• Restricciones del acceso público	Conformidad <ul style="list-style-type: none">• Especificación• Grado de conformidad
Metadatos sobre Metadatos <ul style="list-style-type: none">• Punto de contacto de los metadatos• Fecha de los metadatos• Lengua de los metadatos	Organizaciones Responsables del Establecimiento, Gestión, Mantenimiento y Distribución de los Conjuntos y Servicios de Datos Espaciales <ul style="list-style-type: none">• Parte responsable• Función de la parte responsable

Figura 4: Elementos de metadatos de INSPIRE.

1.3.1.1.4.3. ISO 19115 WISE

Este perfil fue desarrollado por la Comisión Europea en documentos como “WISE GIS Guidance Document” (2008), en la elaboración del mismo participaron los técnicos de la Confederación Hidrográfica del Duero.

En el 2002 se definió en Europa, el primer perfil de metadatos WISE (Water Information System for Europe) basado en la norma ISO 19115. Fue actualizado en el 2008 a partir de investigaciones realizadas por expertos de la Directiva Marco del Agua, específicamente en aquellas que trataban el manejo de Sistemas de Información Geográficos.

Este perfil incluye una serie de elementos del núcleo de la norma ISO 19115 como se muestra en la figura 5.

Elementos obligatorios del Núcleo de ISO	Elementos opcionales y condicionales del Núcleo de ISO
<ul style="list-style-type: none">• Punto de contacto del metadato• Fecha de creación• Título del conjunto de datos• Fecha de referencia del conjunto de datos• Resumen descriptivo• Versión de la Norma de Metadatos• Idioma del conjunto de datos	<ul style="list-style-type: none">• Identificador del Fichero de Metadatos• Norma de Metadatos• Parte responsable del Conjunto de Datos• Tipo de representación espacial• Idioma del metadato• Conjunto de caracteres de los Metadatos• Conjunto de caracteres del Conjunto de Datos• Resolución espacial del conjunto de datos• Categoría del tema• Extensión vertical y temporal• Formato de Distribución

Figura 5: Elementos de WISE.

1.3.2. Ámbito de Cuba

La Oficina Nacional de Recursos Minerales (ONRM) ha desarrollado un perfil de metadatos que está registrado en el documento "Procedimiento para la Documentación del Metadato". En octubre del 2007 la Geominera Oriente, como parte del Programa Geodato, publicó una versión corregida de este procedimiento. Dentro de este programa se realizó el análisis de contenido del metadato para el conjunto de datos geológicos.

Se consideró como conjunto de datos geológicos a la unidad de información que se ha digitalizado u organizado como producto de la ejecución de un proyecto de conformación de base de datos.

(Programa Geodato, 2007)

Este perfil fue realizado siguiendo la norma ISO 19115.TC211, que fue desarrollada en el 2003 por el comité técnico numero 211 de la ISO con el propósito de hacer análogas otras normas de metadatos geológicos que se habían elaborado anteriormente, además de pretender llegar a un acuerdo internacional sobre metadatos geoespaciales.

El perfil cubano de metadato contiene ocho de los catorce paquetes UML que propone la ISO 19115, estos son:

- Información del Metadato.
- Identificación de los Datos.
- Información del Contenido.
- Calidad de los Datos.
- Representación Espacial.
- Sistema de Referencia.
- Distribución de los Datos.
- Información sobre las Citas.

Finalmente, el metadato quedó constituido por este grupo de paquetes interrelacionados, que a su vez contienen uno o varios modelos que agrupan toda la información.

1.3.3. Definiendo la norma a utilizar

Para dar solución al problema que se plantea en la investigación se trazó como objetivo la creación de un módulo para el catálogo de mapas LiberMaps que permita la gestión de metadatos geográficos, este módulo no será creado a la medida de ningún negocio en particular, tiene un carácter genérico con el fin de ser personalizado a un entorno determinado, lo que trae por consiguiente que la norma que se defina para desarrollar este sistema no debe ser específica, sino que pueda ser extrapolada a cualquier ámbito.

La ISO 19115 es la norma apropiada si se quiere trabajar en el campo de los metadatos relacionados con datos y servicios geográficos. La misma constituye la normativa que se sigue en la investigación por ser mundialmente utilizada, además es una norma genérica que cubre la totalidad de los datos necesarios para cualquier recurso que se describa.

En una primera versión del sistema se implementa el núcleo o core de la norma ISO 19115 pues este define el conjunto mínimo de metadatos requeridos para soportar todo el rango de aplicaciones de metadatos (descubrimiento de datos, determinación de la idoneidad de algunos para un uso, acceso a los

datos, transferencia de datos y utilización de datos digitales). En versiones posteriores se le incorporarán otros perfiles y la norma ISO 19115 en su totalidad de acuerdo a las necesidades de las organizaciones o entidades interesadas en el sistema a implementar.

1.4. Situación Problemática

Los catálogos surgen por la necesidad de organizar adecuadamente una determinada información, estos permiten localizar, identificar, valorar y obtener información de un tema en particular, como por ejemplo: ¿Qué existe sobre determinado tema?, ¿Cómo se titula? Pero también existen los catálogos en línea que brindan la posibilidad de consultarlos desde cualquier sitio donde se encuentre el usuario, además tienen la capacidad de guiar a este en una búsqueda de materiales.

En los últimos tiempos se ha observado un amplio crecimiento en la búsqueda de información geográfica tanto por personas como por empresas, casi todos los sucesos o fenómenos de la actualidad tienen una componente geográfica. Cuando se desea conocer sobre una determinada información geográfica, una opción recomendable y útil es buscar en catálogos de mapas.

En la UCI, un grupo de trabajo de la facultad 9 está desarrollando el catálogo de mapas LiberMaps y en el marco de este trabajo se detectó la necesidad de promover la reusabilidad de datos sin tener que recurrir al equipo humano que los creó inicialmente, que la información geográfica quede publicada para que diferentes organizaciones puedan usarla y compartir los datos con otras. También era preciso lograr la interoperabilidad de la información al venir esta de distintos orígenes.

Por todo lo anteriormente expuesto es que surgió la idea de implementar un módulo para la gestión de metadatos para el catálogo LiberMaps. Los metadatos ayudan a documentar los datos lo cual es de gran valor, permiten a los usuarios utilizarlos de forma eficaz y decidir si serán de interés para ellos, además de ayudarlos en el procesamiento, interpretación, y almacenamiento de los datos en repositorios internos.

1.5. Análisis de otras soluciones existentes

Hoy en día se pueden encontrar distintas herramientas que permiten crear metadatos atendiendo a las normas (incluyendo extensiones y perfiles) presentados en los epígrafes anteriores.

Se puede citar al **CatMDEdit** que facilita el manejo de recursos a través de los metadatos asociados a estos, haciendo mayor hincapié en la gestión y documentación de recursos de información geográfica. Es una herramienta de código abierto desarrollada por el consorcio TeIDE³. Fue implementada en Java y unas de sus características más importantes es que es multiplataforma, es decir, que funciona en cualquier sistema operativo (Windows, Linux, Mac y otros.). Permite la creación de metadatos de acuerdo a distintos perfiles de ISO 19115 (XML según el esquema definido por ISO/TS 19139).

Se ha identificado otra herramienta como **ISO Metadato Editor (IME)**, desarrollada por el Instituto Nacional de Técnicas Aeroespaciales (INTA) del Ministerio de Defensa. Admite la definición y modificación de perfiles, la creación de extensiones, edición de metadatos, generación de ficheros XML y la creación y lectura de una plantilla de texto para facilitar la conexión con otros programas. Esta herramienta permite la creación de metadatos de acuerdo a la norma ISO 19115 y brinda la posibilidad de validar archivos XML según el esquema definido por ISO/TS 19139.

También se puede mencionar al **MetaD**, programa gratuito de edición y exportación de metadatos desarrollado por el ICC para dar soporte a la IDE de Cataluña. Esta aplicación persigue como objetivo más importante crear metadatos de forma guiada, de manera que les facilite a los usuarios crearlos, editarlos, mantenerlos y exportarlos. Se trata de un subconjunto del estándar ISO 19115, con su implantación ISO 19139, destinado a describir la información geográfica.

Otra muestra es **GeoNetwork**, la cual es una aplicación gratuita y también de código abierto. Posee un entorno estándar diseñado para permitir el acceso a información georreferenciada, productos cartográficos y metadatos relacionados procedentes de diversas fuentes, exaltando el intercambio de información espacial entre organizaciones y usuarios. Permite la edición en línea de los metadatos con un poderoso sistema de plantillas, la recolección periódica de datos y crear metadatos usando un editor en línea. En esta aplicación también se puede cargar XML, importar o exportar archivos MEF que contienen metadatos. Aplica las normas de metadatos ISO19139 (19115 y 19119), Dublin Core.

La aplicación **ArcCatalog** organiza y maneja toda la información de SIG como los mapas, globos, conjuntos de datos, modelos, metadatos, y servicios. Incluye la búsqueda y hallazgo de información geográfica; el registro, visualización, y manejo de metadatos, definición, exportación, e importación de

³ TeIDE es un consorcio de equipos de I+D de las Universidades de Zaragoza (Dpto. de Informática e Ing. de Sistemas), Jaume I (Dpto. de Lenguajes y Sistemas Informáticos) y Politécnica de Madrid (Dpto. de Ing. Topográfica y Cartografía).

esquemas de geodatabases, la búsqueda y descubrimiento de datos SIG en las redes locales y la Web. Los usuarios utilizan ArcCatalog para organizar, encontrar, y manipular los datos SIG, así como para obtener datos que usan los metadatos basados en normas establecidas. Fue desarrollada en una versión atrasada de la norma ISO 19115, por lo que los metadatos creados no son compatibles con otras herramientas que utilizan la versión actualizada del estándar.

Después de haber realizado un análisis de las principales características de varias herramientas utilizadas en la creación de metadatos y otras funciones relacionadas con estos, se decide descartar varias de estas herramientas debido a su condición de software propietario como es el caso de ArcCatalog, y basar la implementación del sistema gestor de metadatos en la herramienta mencionada anteriormente: CatMDEdit, pues es una de las más completas, ya que implementa las principales normas que existen para la creación de metadatos geográficos, y de fácil entendimiento, además de ser de código abierto; es una de las herramientas más poderosas en el proceso de captación del metadato, es decir, su creación. También tuvo un gran peso en esta decisión la opinión de los especialistas funcionales del grupo de trabajo de la plataforma GeneSIG. Se recomienda para el proceso de publicación utilizar Geonetwork, por lo que en otra versión del sistema se implementará un componente para la publicación guiado por esta herramienta (Geonetwork).

1.6. Conclusiones

Durante el desarrollo de este capítulo se analizaron primeramente varios conceptos que apoyan al entendimiento de la investigación, y además las distintas normas que existen para la creación de metadatos de propósito general y geográfico. Luego de realizar el análisis del estado del arte y estudiar detenidamente la situación problemática planteada, se definen primeramente, la norma a utilizar en la construcción del sistema gestor de metadatos, consistente en la ISO 19115 Core, pues esta define el conjunto mínimo de metadatos requeridos para soportar todo el rango de aplicaciones de los mismos, y después de analizar varias aplicaciones de este tipo se expuso también que la implementación de esta solución se basaría en la herramienta CatMDEdit, ya que es una de las más completas en el proceso de creación de los metadatos. Estos aspectos proveen la base de conocimientos necesaria para llevar a cabo la implementación del sistema.

CAPITULO 2. Tendencias y tecnologías actuales a utilizar.

2.1 Introducción.

En este capítulo se hace un estudio de metodologías de desarrollo de software que facilitan un mejor entendimiento de todo el flujo de trabajo. También se caracterizan una serie de tecnologías y herramientas para la construcción de software existentes en la actualidad. Se realiza una comparación entre más de una de ellas arribando a conclusiones sobre la que se utiliza para la elaboración del sistema propuesto.

2.2 Metodologías de Desarrollo de Software.

Las metodologías y estándares utilizados en el desarrollo de software proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado. **(CHACÓN, 2006)**

En las últimas décadas se ha incrementado notablemente el interés en la realización de productos de software, y para la realización de estos existen dos tipos de metodologías de desarrollo de software: Metodologías Ágiles y Metodologías Tradicionales o Robustas. Las primeras se caracterizan por permitir a los equipos desarrollar software rápidamente y responder a los cambios que puedan surgir a lo largo del proyecto, pero también presentan algunos inconvenientes ya que estas metodologías están dirigidas específicamente a equipos pequeños o medianos que posean un ambiente donde predomine la comunicación y colaboración entre todos los miembros del equipo durante todo el tiempo, y cualquier resistencia del cliente o del equipo de desarrollo hacia las prácticas y principios puede llevar el proceso al fracaso. Las segundas en cambio se caracterizan por centrarse especialmente en el control del proceso con el fin de conseguir un software más eficiente y, aunque posee una cierta resistencia a los cambios, está dirigido a grandes grupos de trabajo.

Entre las metodologías más utilizadas de los últimos tiempos se encuentra Rational Unified Process (RUP), de la que se abordan algunas características en este epígrafe para poder argumentar así la selección realizada.

2.2.1 Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, proporciona disciplinas en las cuales se encuentran artefactos con lo cual se podrá contar con guías para poder documentar e implementar de una manera fácil y eficiente, todas las guías para un buen desarrollo, todo esto dentro de las respectivas fases con las cuales cuenta. **(CHACÓN, 2006)**

Según RUP, el desarrollo del software se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición, cada una de las cuales es desarrollada mediante ciclos de iteraciones.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales como se muestra en la figura 6. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. **(CHACÓN, 2006)**

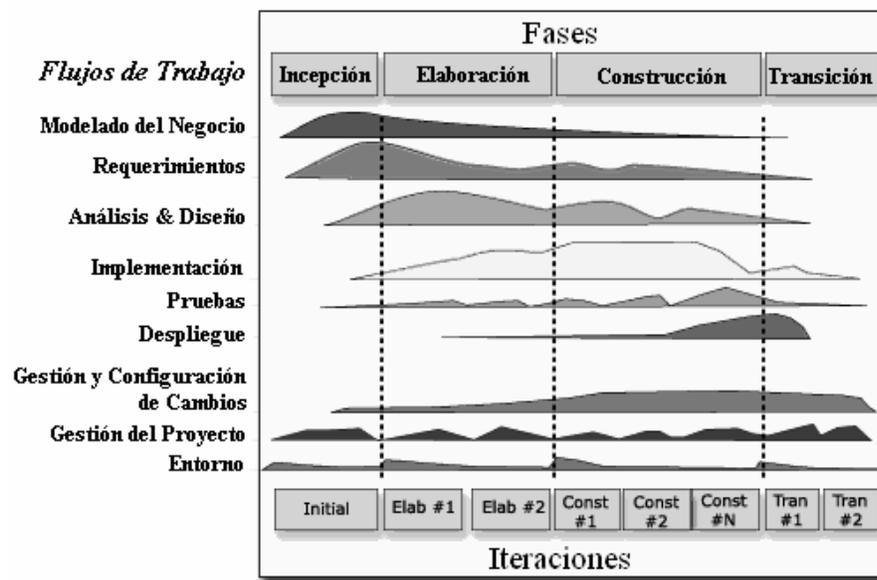


Figura 6: Disciplinas, Fases e Iteraciones de la Metodología RUP.

RUP es un proceso que está definido por tres características esenciales: dirigido por los casos de uso, iterativo e incremental y centrado en la arquitectura.

2.3 Lenguaje Unificado de Modelado (UML).

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. **(Popkin Software and Systems, 2008)**

Sus principales funciones son las de Visualizar, Especificar, Construir y Documentar. A través de este lenguaje se puede modelar, lo mismo sistemas de software, hardware u organizaciones del mundo real.

Un modelo UML está compuesto por tres clases de bloques de construcción: Elementos, Relaciones y Diagramas. **(Orallo, 2007)**

2.4 Servidores Web.

Un servidor Web es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Language). La comunicación de estos datos entre cliente y servidor se hace por medio de un protocolo, concretamente del protocolo HTTP. El servidor es el responsable de proporcionar el acceso a los recursos solicitados que están bajo el control del sistema operativo. Estos recursos pueden ser estáticos, como páginas HTML o texto y dinámicos, como por ejemplo CGI's. (Common Gateway Interface o Pasarela de Interface Común). **(Fernández, 2008)**

2.4.1 Servidor HTTP Apache.

El Apache HTTP Server Project es un esfuerzo para desarrollar y mantener una fuente abierta, servidor HTTP para sistemas operativos modernos, incluyendo UNIX y Windows NT. El objetivo de este proyecto es proporcionar un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sintonía con los estándares HTTP actuales. El servidor HTTP Apache es un proyecto de la Fundación de Software Apache.

Apache Server permite la creación y publicación de documentos php de la misma forma que se hace en Internet con una estabilidad y eficacia ampliamente comprobada en la gran cantidad de servidores apache actualmente en uso. Este servidor es uno de los mayores triunfos del software libre.

Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Implementa los últimos protocolos, aunque se base en el HTTP / 1.1

Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.

Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Es un servidor altamente configurable de diseño modular ya que es muy sencillo ampliar las capacidades del mismo. Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor, esto es posible al configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Presenta un alto nivel de configuración en la creación y gestión de logs pues permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. **(Apache, 2007)**

2.5 Lenguajes de Programación Web.

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de términos monosémicos (es decir, con sentido único) y una regla principal que resume las demás, todas estas son utilizadas por el programador a través de las cuales crea un programa o subprograma. **(Enciclopedia Libre Universal en Español)** En este epígrafe se abordan algunas características de lenguajes de programación del lado del servidor y del lado del cliente, para poder argumentar con estos elementos los lenguajes seleccionados.

2.5.1 Lenguajes del lado del servidor.

PHP

PHP es un lenguaje script procesado en el lado del servidor. PHP se muestra como código embebido dentro de una página HTML. El funcionamiento de PHP se evidencia cuando el Navegador realiza una

petición al servidor (se escribe la URL), después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador y por último el Navegador muestra la respuesta del servidor.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se pueda realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. **(Rodas, 2007)**

2.5.2 Programación del lado del cliente.

ExtJs

ExtJS es un framework JavaScript para interfaces avanzadas. Con el tiempo y la experiencia, se ha mejorado y desarrollado, integrándole un sin número de librerías para infinidad de problemas al programar en la web. Entre las principales actividades que se hacen se encuentran: mensajes de advertencia, listas

desplegables, validación de campos en formularios, entre otras. Con su uso, se hace más fácil la programación. ExtJS constituye un marco de **JavaScript** con **AJAX** y componentes de interfaz de usuario. **(Sencha Solutions, 2008)**

Javascript es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. **(Angel, 2006)**

El término **AJAX** es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". Ajax se trata de varias tecnologías independientes que se unen de forma novedosa. Las tecnologías que forman AJAX son: XHTML y CSS, para crear una presentación basada en estándares; DOM, para la interacción y manipulación dinámica de la presentación; XML, XSLT y JSON, para el intercambio y la manipulación de información; XMLHttpRequest, para el intercambio asíncrono de información; JavaScript, para unir todas las demás tecnologías. **(Eguíluz, 2006)**

ExtJS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, ExtJS no se preocupará de lo que pase en el servidor. Hay docenas de widgets a escoger en ExtJS, incluyendo composiciones automáticas de páginas, pestañas, menús, barras de herramientas, diálogos, vistas en árbol. Proporciona un selector de nodos DOM extremadamente poderoso llamado DomQuery (puede usarse como una librería independiente, pero en el contexto de ExtJS se usará para seleccionar elementos para poder interactuar con ellos a través de la interfaz Element; contiene muchos de los métodos y propiedades de DOM que se necesitará, proporcionando una interfaz conveniente, unificada y multinavegador). **(Sencha Solutions, 2008)**

2.6 Herramientas.

Las herramientas son aplicaciones empleadas para la construcción de otros programas o aplicaciones. Estas mejoran la calidad y la productividad de los sistemas. Tanto el proceso de desarrollo de sistemas como el producto que se obtiene con él, pueden mejorarse con el uso de herramientas apropiadas.

Con las herramientas correctas, se tiene el potencial de ser más productivo; se pueden completar las mismas actividades de desarrollo en un tiempo menor que el que se utiliza cuando no se hace uso de las

herramientas. En algunos casos, las herramientas correctas ayudan a alcanzar un nivel de productividad que hace factible una tarea que de otro modo no sería posible realizar. Las herramientas pueden sugerir la mejor forma para abordar una tarea. **(Ochoa, 2005)** A continuación se abordan las características de varias herramientas para ayudar en la selección de las más apropiadas para la construcción de la solución propuesta.

2.6.1 Sistemas Gestores de Bases de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) es un software que proporciona servicios para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en base de datos de forma segura y eficiente. Un SGBD actúa como un intermediario entre las aplicaciones y los datos. **(Sicilia, 2008)**

Entre los más utilizados mundialmente se encuentra PostgreSQL, del cual se aborda en este epígrafe lo que ayudó a seleccionarlo para emplearse en la construcción del sistema propuesto.

2.6.1.2 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD⁴ y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

A continuación se exponen algunas de las características más importantes de PostgreSQL:

⁴ La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva. La licencia BSD permite el uso del código fuente en software no libre.

Generales

- Es una base de datos 100% ACID⁵
- Integridad referencial
- Tablespaces
- Replicación asíncrona
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Multi-Version Concurrency Control (MVCC)
- Acceso encriptado vía SSL
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows **(Martinez, 2009)**

PostGIS

El PostGIS es un módulo que añade entidades geográficas al PostgreSQL. Nativamente, el PostgreSQL ya soporta geometrías espaciales, sin embargo, el PostGIS añade la capacidad de almacenamiento/recuperación según la especificación SFS (Simple Features Specification) del consorcio internacional Open GeoSpatial (OGC). Además del almacenamiento de datos geográficos, este módulo también implementa diversas funcionalidades topológicas, posibilitando el desarrollo de SIG Corporativos. La topología también forma parte de la especificación SFS (OpenGIS®), garantizando al PostGIS interoperabilidad con incontables sistemas que también adoptan el SFS.

⁵ ACID: Atomicidad, Consistencia, Aislamiento y Durabilidad.

El licenciamiento del PostGIS es definido por la GNU GPL (General Public License), garantizando todas las libertades de un software libre. Fue desarrollado por la empresa canadiense Refractions Research. **(Nogueira, y otros, 2005)**

2.6.2 Herramientas CASE.

Una herramienta CASE es un software que se utiliza para ayudar a las actividades del proceso de software, o software que es utilizado para diseñar y para implementar otro software. Facilitan mejoras en la calidad y productividad del diseño y desarrollo. Ayudan a automatizar metodologías de software y desarrollo de aplicaciones.

Estas herramientas proporcionan a los desarrolladores de software grandes ventajas: apoyan a las metodologías y métodos, mejoran la comunicación entre las personas que interactúan con el sistema permitiéndoles compartir mejor su trabajo, establecen métodos eficientes para almacenar y utilizar los datos, hacen más eficaz el mantenimiento y automatizan fragmentos del análisis y diseño pesados y vulnerables a errores.

A continuación se abordan las características de una de estas herramientas para así argumentar su selección de manera tal que se garantice una mayor eficiencia en el desarrollo del software.

2.6.2.2 Visual Paradigm.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Características

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio.

- Documento Modelado colaborativo con CVS y Subversion
- Interoperabilidad con modelos UML2
- Código a modelo.
- Código a diagrama
- Ingeniería inversa Java, C++.
- Generación de código
- Modelo a código, diagrama a código
- Editor de Detalles de Casos de Uso.
- Diagramas de flujo de datos
- Generador de informes para generación de documentación
- Distribución automática de diagramas (**Visual Paradigm International, 2010**)

2.6.3 Framework.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (**Potencier, y otros, 2008**)

2.6.3.2 Framework Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El

resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web.

Características de Symfony:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “*convenir en vez de configurar*”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de *mejores prácticas* y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Este framework está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. **(Potencier, y otros, 2008)**

2.6.4 Entorno Integrado de Desarrollo (IDE).

Un entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse a un sólo lenguaje de programación o a varios.

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes, incluso en algunos de ellos pueden funcionar como un sistema en tiempo de ejecución donde se puede emplear el lenguaje de programación en forma interactiva.

A continuación se exponen las características fundamentales del IDE NetBeans, el cual se emplea en la construcción del sistema propuesto.

2.6.4.2 NetBeans

El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso.

Es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. Soporta el desarrollo de todos los tipos de aplicación Java. Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactoring.

La versión actual es NetBeans IDE fue lanzada en Mayo de 2007. Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios, y modelado UML.

Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente. **(Wheeler, 2008)**

2.6 Justificación de la selección.

Luego de haber analizado metodologías, tecnologías, lenguajes de programación y herramientas utilizadas en el desarrollo de software se realizó una selección de estas para desarrollar la solución propuesta, para ello se tuvo en cuenta como principal elemento a considerar que fueran libres o de código abierto, pues en Cuba actualmente se aboga por la soberanía e independencia tecnológica y la UCI no se encuentra exenta de esto. También se consideró que esta solución es un módulo que integra al catálogo LiberMaps que a su vez forma parte de la plataforma GeneSIG de la cual ya fueron definidas las tecnologías por los especialistas del equipo de trabajo, por lo tanto en la decisión a tomar se debe ser consecuente con el SGBD, la programación del lado del cliente, Servidor Web, metodología, herramienta case, entre otras tecnologías.

A partir de lo planteado se seleccionó como lenguaje de programación del lado del servidor: PHP, por su rapidez, su facilidad de aprendizaje, su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos, y el hecho de que se distribuye de forma gratuita bajo una licencia de código abierto.

Del lado del cliente se propone utilizar el framework ExtJs para el desarrollo de las interfaces pues además de permitir la reutilización de código, es independiente o adaptable a frameworks diferentes y proporciona una interfaz conveniente, unificada y multinavegador.

Se ha seleccionado además la herramienta NetBeans como editor de php, pues ofrece un eficiente auto-completado de código, marcado de error de PHP, plantillas, macros, control de versiones, también es muy importante decir que es multiplataforma y de código abierto lo que se corresponde con el objetivo principal de este proyecto.

Como framework a utilizar se escogió Symfony, ya que permite un rápido desarrollo de web construidas con técnicas probadas y estándar, menos errores debido al uso de código fuente ampliamente probado y menos líneas de código. Las aplicaciones Web resultantes son más sencillas de mantener y/o ampliar debido al uso del patrón Modelo Vista Controlador (MVC), que divide la aplicación en capas separadas, y por último, pero no menos importante, es que es publicado bajo licencia de código abierto.

El servidor Web elegido es Apache, pues brinda varias ventajas dentro de las que se encuentra fundamentalmente que es de código abierto, otros aspectos muy importantes que lo caracterizan son su estructura modular, además de ser multiplataforma, extensible y la facilidad de encontrar ayuda y soporte para el mismo.

Como sistema de gestión de base de datos, se seleccionó PostgreSQL, el cual también es multiplataforma y de código abierto, además de ser muy estable y flexible, de alto rendimiento y extensible. También se destaca por ser diseñado para ambientes de alto volumen, además soporta geometrías espaciales y contiene el módulo postgis que añade la capacidad de almacenamiento/recuperación según la especificación SFS (Simple Features Specification) del consorcio internacional Open GeoSpatial (OGC), lo cual es ideal para esta solución.

En epígrafes anteriores se explicaba la importancia que tiene escoger la metodología adecuada para el desarrollo de un software. En este caso, luego de haber analizado las metodologías RUP y XP se decidió utilizar RUP pues aunque parezca un proyecto a corto plazo se necesita de una completa documentación del mismo, y esta metodología tiene la particularidad de que en cada ciclo de iteración, se hace exigente el uso de artefactos, de esta forma, al finalizar el proyecto se contará con una exhaustiva información del mismo.

Como herramienta de modelado CASE se utiliza Visual Paradigm en primer lugar porque es una herramienta que puede ser ejecutada sobre entornos libres, de la cual se ha ido adoptando su utilización en la UCI precisamente por esta razón. Además, tiene una buena integración con IDEs, es muy personalizable y soporta UML 2.1 y muchos lenguajes de programación.

2.7 Conclusiones.

En este capítulo se ha realizado una comparación entre varias tecnologías y herramientas, de esta forma se ha fundamentado la selección de estas para el desarrollo de la solución propuesta. Se seleccionaron los framework ExtJs del lado del cliente, y Symfony del lado del servidor, en los cuales se utilizan como lenguaje de programación, javascript y php respectivamente. Para llevar a cabo la implementación del sistema se seleccionó el IDE NetBeans en su versión 6.8, el sistema de gestión de base de datos PostgreSQL en su versión 8.2 y Apache como servidor Web. La metodología que guiará el proceso de desarrollo será RUP que emplea como lenguaje de modelado UML y se utilizará la herramienta CASE: Visual Paradigm. El correcto empleo de estas tecnologías y herramientas contribuye a obtener un software con calidad. Después de haber hecho una completa selección de las tecnologías y herramientas para desarrollar el presente proyecto, se está en condiciones de realizar una presentación de la solución propuesta.

CAPÍTULO 3: Presentación de la solución propuesta.

3.1. Introducción

En este capítulo se comienza la construcción del sistema Módulo para la gestión de metadatos geográficos de LiberMaps generándose todos los artefactos correspondientes siguiendo la metodología de desarrollo de software propuesta: RUP. Se mostrarán los principales conceptos del entorno a través del modelo de dominio así como la descripción textual de cada uno de estos conceptos. También se especifican los requisitos funcionales (RF) y no funcionales (RNF) que el sistema debe cumplir y se representa una muestra de la solución propuesta mediante el diagrama de casos de uso del sistema, las descripciones detalladas del actor y de uno de los casos de uso.

3.2 Modelo de dominio

El módulo para la gestión de metadatos geográficos de LiberMaps que se va a desarrollar ha sido concebido como un producto para comercializar a cualquier cliente o entidad interesada. Debido a esto se hace difícil definir procesos de negocio bien estructurados que permitan construir un modelo de negocio completo y por lo tanto se decidió realizar un modelo de dominio.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. **(Jacobson, y otros, 2000)**

El objetivo fundamental que se persigue al realizar el modelo de dominio es comprender y describir las clases más importantes dentro del contexto del sistema. Además, este ayuda a los usuarios a mantener una terminología común lo que les permite compartir el conocimiento de forma más consistente.

3.2.1. Diagrama de clases del Modelo de Dominio

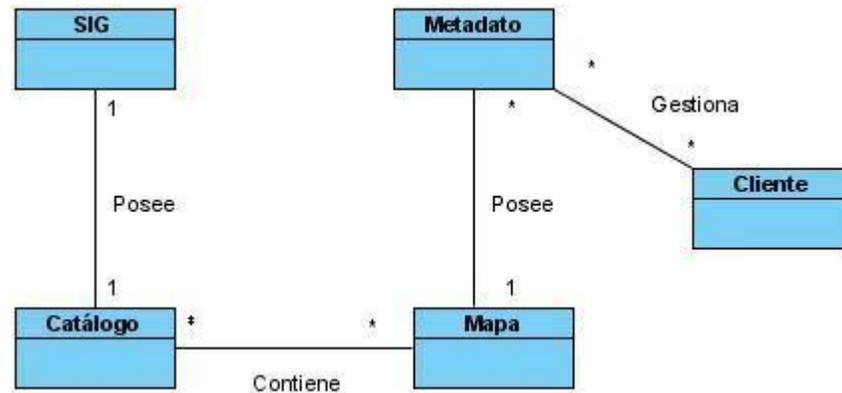


Figura 7: Diagrama de Clases del Modelo de Dominio.

3.2.2. Glosario de Términos del Dominio

SIG: Es el “conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. Los SIG son una tecnología que permite gestionar y analizar la información espacial, y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato”. (RODRÍGUEZ *et al.*, 1998).

Catálogo: Es la lista ordenada o clasificada que se hará sobre cualquier tipo de objetos (monedas, bienes a la venta, documentos, entre otros) o en su defecto personas.

Mapa: Es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.

Metadatos: Los metadatos se definen comúnmente como "datos acerca de los datos". Describen el contenido, la calidad, el formato y otras características que llevan asociadas un recurso, constituyendo un mecanismo para caracterizar datos y servicios de forma que usuarios (y aplicaciones) puedan localizarlos y acceder a ellos.

Cliente: Persona, Organismo u Organización que necesite acceder a los mapas mediante sus metadatos.

3.3. Modelo del Sistema

A continuación se presentarán los artefactos fundamentales del flujo de trabajo: Requisitos, enmarcado en la Fase de Inicio, se definirán los requisitos funcionales y no funcionales, se describirán los actores y los casos de uso del sistema.

3.3.1. Requisitos Funcionales

Un requisito se puede definir como **una condición o capacidad** que un usuario necesita para resolver un problema o lograr un objetivo.

El propósito fundamental de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos funcionales, que son condiciones o capacidades que el sistema debe cumplir, suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre que debe y que no debe hacer el sistema. **(Jacobson, y otros, 2000)**

A continuación se exponen los requisitos funcionales que debe cumplir el sistema que se propone.

El sistema debe permitir:

RF1. Editar Datos Generales de cada metadato.

RF2. Gestionar Información del Responsable de cada metadato.

RF2.1. Registrar Información del Responsable de cada metadato.

RF2.2. Editar Información del Responsable de cada metadato.

RF2.3. Eliminar Información del Responsable de cada metadato.

RF3. Gestionar Información de Contacto de los responsables de cada metadato.

RF3.1. Registrar Información de Contacto de los responsables de cada metadato.

RF3.2. Editar Información de Contacto de los responsables de cada metadato.

RF3.3. Eliminar Información de Contacto de los responsables de cada metadato.

RF4. Gestionar Información del Sistema de Referencia de cada metadato.

RF4.1. Registrar Información del Sistema de Referencia de cada metadato.

RF4.2. Editar Información del Sistema de Referencia de cada metadato.

RF4.3. Eliminar Información del Sistema de Referencia de cada metadato.

RF5. Gestionar Información de las Menciones.

RF5.1. Registrar Información de las Menciones.

RF5.2. Editar Información de las Menciones.

RF5.3. Eliminar Información de las Menciones.

RF6. Gestionar Información de Identificación de Datos de cada metadato.

RF6.1. Registrar Información de Identificación de Datos de cada metadato.

RF6.2. Editar Información de Identificación de Datos de cada metadato.

RF6.3. Eliminar Información de Identificación de Datos de cada metadato.

RF7. Gestionar Información de RS_Identificador.

RF7.1. Registrar Información de RS_Identificador.

RF7.2. Editar Información de RS_Identificador.

RF7.3. Eliminar Información de RS_Identificador.

RF8. Gestionar Información de Extensión de Elemento Geográfico.

RF8.1. Registrar Información de Extensión de Elemento Geográfico.

RF8.2. Editar Información de Extensión de Elemento Geográfico.

RF8.3. Eliminar Información de Extensión de Elemento Geográfico.

RF9. Gestionar Información de Extensión de Elemento Temporal.

RF9.1. Registrar Información de Extensión de Elemento Temporal.

RF9.2. Editar Información de Extensión de Elemento Temporal.

RF9.3. Eliminar Información de Extensión de Elemento Temporal.

RF10. Gestionar Información de Extensión de Elemento Vertical.

RF10.1. Registrar Información de Extensión de Elemento Vertical.

RF10.2. Editar Información de Extensión de Elemento Vertical.

RF10.3. Eliminar Información de Extensión de Elemento Vertical.

RF11. Gestionar Información de Distribución de cada metadato.

RF11.1. Registrar Información de Distribución de cada metadato.

RF11.2. Editar Información de Distribución de cada metadato.

RF11.3. Eliminar Información de Distribución de cada metadato.

RF12. Gestionar Información sobre Calidad de los Datos de cada metadato.

RF12.1. Registrar Información sobre Calidad de los Datos de cada metadato.

RF12.2. Editar Información sobre Calidad de los Datos de cada metadato.

RF12.3. Eliminar Información sobre Calidad de los Datos de cada metadato.

RF13. Exportar archivo XML de Metadatos.

RF14. Importar archivo de Metadatos.

RF15. Gestionar Repositorio.

RF15.1. Registrar Repositorio.

RF15.2. Editar Repositorio.

RF15.3. Eliminar Repositorio.

RF16. Visualizar Metadatos.

RF17. Gestionar Metadatos.

RF17.1. Añadir Metadatos.

RF17.2. Eliminar Metadatos.

RF18. Replicar Metadatos.

RF19. Gestionar Información de MD_Identificador.

RF19.1. Registrar Información de MD_Identificador.

RF19.2. Editar Información de MD_Identificador.

RF19.3. Eliminar Información de MD_Identificador.

3.3.2. Requisitos no Funcionales

La eficiencia y calidad de la solución propuesta no sólo depende de las capacidades o condiciones que debe cumplir sino que también procura ser más atractivo al cliente, más seguro y rápido. Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

A continuación se exponen los requisitos no funcionales que el sistema propuesto debe cumplir.

Requisitos de Usabilidad

RNF1. El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

RNF2. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.

Requisitos de Fiabilidad

RNF3. El sistema debe estar disponible todo el tiempo para sus usuarios, descontando el tiempo en que se encuentre en mantenimiento.

RNF4. El tiempo entre fallos no debe exceder los 3 meses.

RNF5. El tiempo medio de reparación en caso de fallos es de 7 días.

Requisitos de Eficiencia

RNF6. El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento.

RNF7. Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

Restricciones de diseño

RNF8. El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.

RNF9. Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación).

Requisitos para la documentación de usuarios en línea y ayuda del sistema.

RNF10. El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

Requisitos de Interfaz de usuario

RNF11. El sistema debe tener una apariencia profesional y un diseño gráfico sencillo.

RNF 12. El sistema debe ser intuitivo.

Requisitos de Hardware

Para las PCs clientes:

RNF 13. Se requiere tengan tarjeta de red.

RNF 14. Al menos 64 MB de memoria RAM.

RNF 15. Se requiere al menos 100 MB de disco duro.

RNF16. Procesador 512 MHz como mínimo.

Para los servidores:

RNF17. Se requiere tarjeta de red.

RNF18. El Servidor de Mapas tenga como mínimo 2GB de RAM y 2GB de disco duro.

RNF19. El Servidor de BD tenga como mínimo 2GB de RAM y 10GB de disco duro.

RNF20. Procesador 3 GHz como mínimo.

Requisitos de Software

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto, el servidor del usuario final debe tener como requisitos mínimos de software:

Para las PCs clientes:

RNF21. Se requiere un Navegador como Mozilla Firefox 3.5.5 u otro navegador que cumpla con los estándares W3C.

RNF22. Se requiere un Sistema operativo: GNU/Linux, Windows o Mac OS.

Para los Servidores:

RNF23. Se requiere Sistemas operativos GNU/Linux o Windows Server 2000 o superior.

RNF24. Se requiere un Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.

RNF25. Se requiere PostgreSQL 8.2 como Sistema Gestor de Base de Datos.

RNF26. Se requiere PostGis como extensión de PostgreSQL como soporte de datos espaciales.

Requisitos de Licencia

RNF27. De acuerdo a los tipos de licencias de los componentes y herramientas que se proponen a utilizar para el desarrollo del Sistema de Gestión de Metadatos Geográficos de LiberMaps se puede catalogar legalmente esta arquitectura de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

Requisitos Legales, de Derecho de Autor y otros.

RNF28. El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

RNF29. La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

RNF30. Como producto, el Sistema de Gestión de Metadatos Geográficos de LiberMaps se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

Estándares Aplicables

RNF31. El sistema será desarrollado bajo estándares internacionales como la normativa ISO19115.

3.3.3. Diagrama de Casos de Uso del Sistema (DCUS).

Al definirse los requisitos con que debe cumplir el sistema, es preciso en función de sus características, y siempre apoyados en los patrones existentes para este propósito, concebir el DCUS.

Un DCUS muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Utilizando los patrones CRUD Total, CRUD Parcial y Concordancia Rehusó, se definen 19 casos de uso, relacionados con el actor "Usuario", como se muestra en la figura 8.

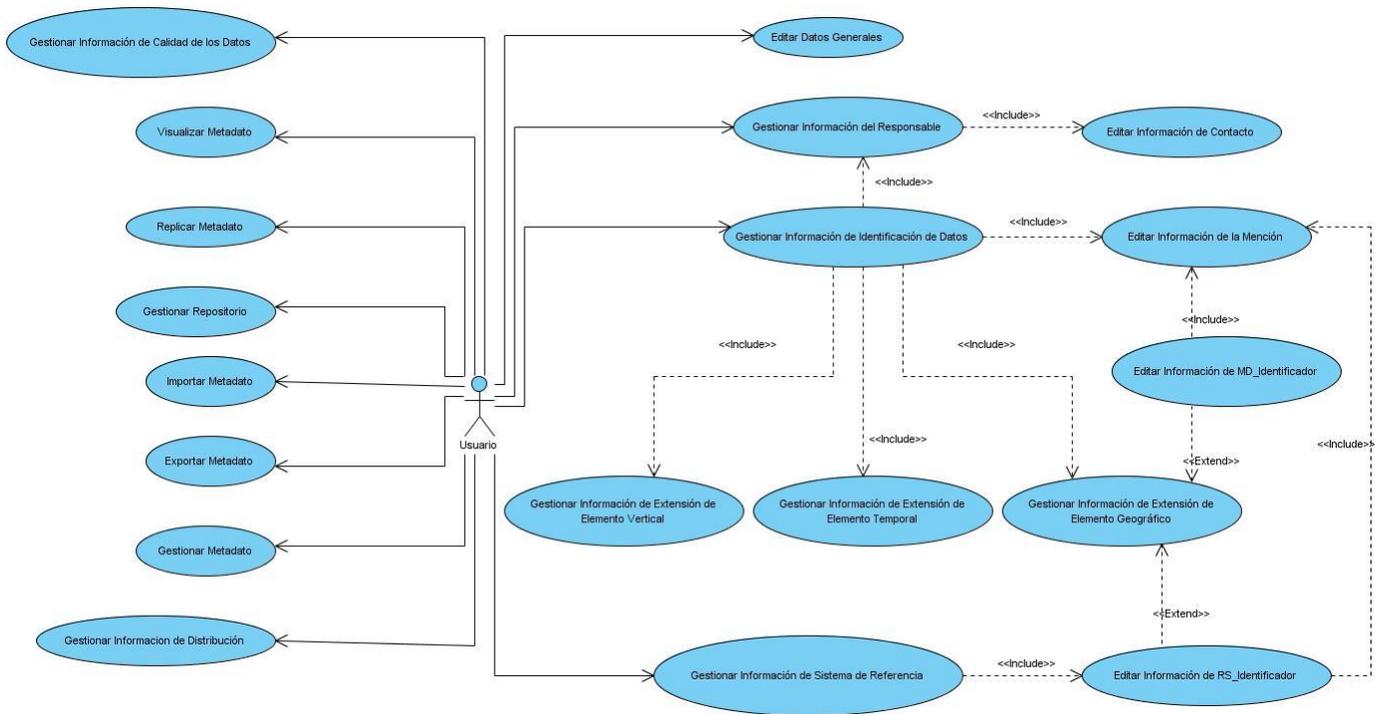


Figura 8: Diagrama de Casos de Uso del Sistema.

3.3.4. Descripción de los actores del sistema.

Un actor representa un comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema.

Actor	Descripción
-------	-------------

Usuario	Responsable de la creación de metadatos y de la correcta documentación de los mismos, a partir de los aspectos que conforman el núcleo de la norma ISO19115. Puede registrar y editar cada uno de estos aspectos; exportar, importar, guardar, añadir, eliminar, visualizar, replicar y editar Metadatos, así como cancelar la edición de los mismos, validar los datos y crear repositorios.
---------	---

Tabla 1: Actor del Sistema.

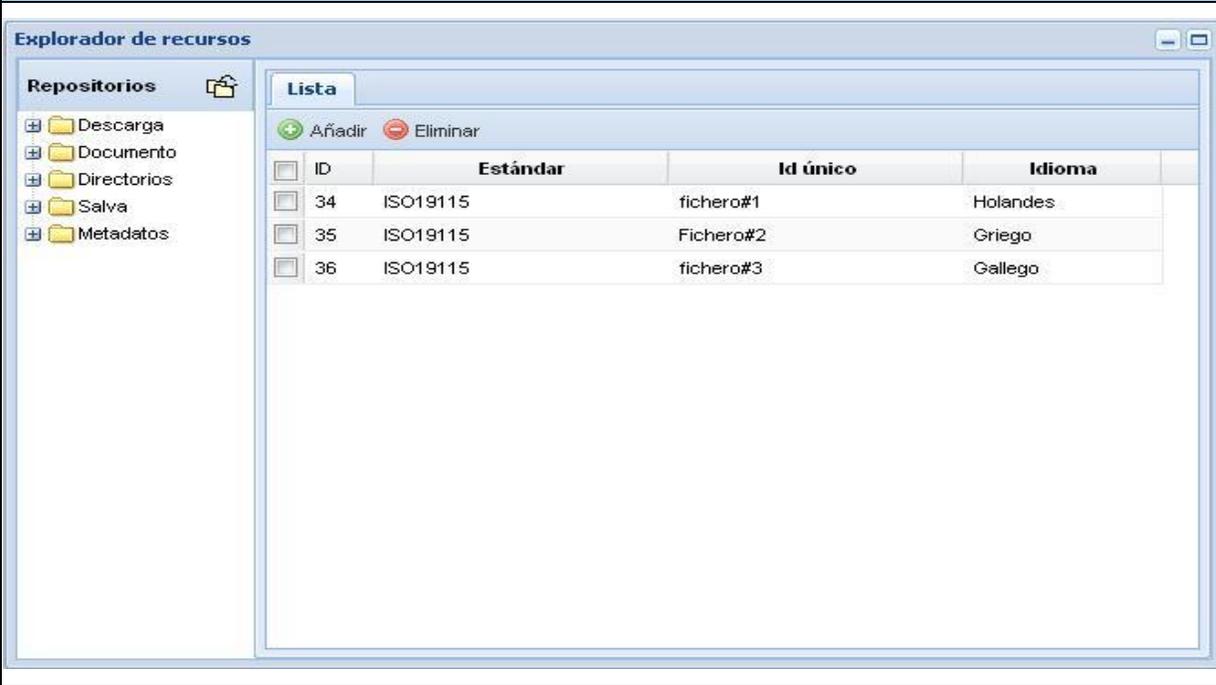
3.3.5. Descripción de los Casos de uso del Sistema.

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad.

A continuación se presenta la descripción del Caso de Uso Gestionar Metadato.

Caso de Uso:	Gestionar Metadato
Actores:	Usuario
Resumen:	El usuario decide añadir o eliminar un metadato. Si decide “añadir”, el sistema inserta un nuevo registro de metadato. Si decide “eliminar”, el sistema elimina el/los registro(s) de metadato(s) seleccionado(s).
Precondiciones:	En el caso de eliminar el usuario debe haber seleccionado el registro de metadato que desea eliminar.
Referencia:	RF17, RF17.1, RF17.2
Prioridad:	Crítico
Flujo Normal de Eventos	
“Gestionar metadato”	
Acción del Actor	Respuesta del Sistema

1. Solicita gestionar un Metadato	2. El sistema muestra un formulario con los repositorios existentes para que el usuario seleccione en cual de ellos desea añadir o eliminar metadatos. (Ver Interfaz 1).
3. Selecciona un repositorio.	4. El sistema muestra un formulario con los metadatos existentes y ofrece opciones "Añadir" o "Eliminar".
5. a) Escoge opción Añadir haciendo clic sobre el botón "Añadir". <i>Sección 1.</i> b) Escoge opción Eliminar seleccionando el/los metadato(s) que desea eliminar y haciendo clic sobre el botón "Eliminar". <i>Sección 3.</i>	
Sección 1: Añadir Metadato	
Acción del Actor	Respuesta del Sistema
	6. El sistema ofrece el formulario de registro de metadato. (Ver Interfaz 2)
7. Introduce los datos en el formulario.	8. Verifica que todos los campos del formulario hayan sido llenados.
	9. Crea el metadato y lo inserta.
Flujo Alternativo de los Eventos	
	9. Encuentra campos vacíos y muestra el mensaje: "Formulario no Válido".
Sección 2: Eliminar Metadato	
Acción del Actor	Respuesta del Sistema

	4. Verifica que haya al menos un metadato seleccionado.
	5. Elimina los metadatos seleccionados y a su vez toda la información asociada a él.
Flujo Alternativo de los Eventos	
	5. Si no hay ningún metadato seleccionado, el sistema muestra el mensaje: "Debe elegir al menos un metadato".
Prototipo de Interfaz	
Interfaz 1	
	
Interfaz 2	

Poscondiciones	-
-----------------------	---

Tabla 2: Descripción del Caso de Uso: Gestionar Metadato.

3.4 Conclusiones

Como resultado de este capítulo se generaron los principales artefactos de los primeros flujos de trabajo propuestos por RUP: Modelamiento del Negocio y Requisitos. Debido a que no pudieron ser identificados procesos de negocio bien definidos, se realizó modelo de dominio, el cual posibilitó un mejor entendimiento del entorno del sistema. Se realizó un levantamiento de requisitos obteniendo 18 requisitos funcionales y 32 no funcionales, se modeló el Diagrama de Casos de Uso del Sistema describiendo al Actor y Casos de Uso que formaron parte de este, todo esto permite contar con una documentación organizada de las restricciones requeridas completándose la Vista de Casos de Uso. De esta forma queda presentada la solución propuesta y se crean las condiciones para pasar a la fase de construcción de la misma.

CAPÍTULO 4: Construcción de la solución propuesta.

4.1. Introducción

En este capítulo se presenta una muestra de los principales artefactos del diseño de la solución propuesta como son el diagrama de clases del diseño del caso de uso Gestionar Metadato, el modelo de datos, el modelo de despliegue y el modelo de implementación. Teniendo en cuenta los frameworks definidos para implementar la aplicación se explica brevemente su funcionamiento para un mejor entendimiento de la solución, así como los patrones de diseño que se utilizan en la construcción de la misma.

4.2. Diseño

El modelo de diseño es el modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen un impacto en el sistema a considerar. **(35)** También el modelo de diseño sirve de abstracción de la implementación del sistema. Este está muy ligado al lenguaje de programación que se emplee por lo que a continuación se realiza una descripción general del funcionamiento de los frameworks utilizados: Symfony y ExtJs, para un mejor entendimiento del flujo descrito en los diagramas de diseño.

4.2.1. Descripción general del funcionamiento del Symfony y ExtJs.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC), que está formado por tres niveles: el modelo, la vista y el controlador. Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

En Symfony, la capa del controlador, que contiene el código que liga la lógica de negocio con la presentación, está integrada por los componentes controlador frontal y las acciones. El controlador frontal es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.

Las acciones contienen la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación.

El controlador frontal es un componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática. Las acciones son el corazón de la aplicación, puesto que contienen toda la lógica de la aplicación. Estas utilizan el modelo y definen variables para la vista. Cuando se realiza una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición. Son métodos con el nombre `executeNombreAccion` de una clase llamada `nombreModuloActions` que hereda de la clase `sfActions` y se encuentran agrupadas por módulos. La clase que representa las acciones de un módulo se encuentra en el archivo `actions.class.php`, en el directorio `actions/` del módulo.

Las clases de la capa del modelo también se generan automáticamente, en función de la estructura de datos de la aplicación. El componente que se encarga por defecto de gestionar el modelo en Symfony es una capa de tipo ORM (*object/relational mapping*) realizada mediante el proyecto Propel. En las aplicaciones Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realiza mediante objetos; de esta forma nunca se accede de forma explícita a la base de datos. Este comportamiento permite un alto nivel de abstracción y una fácil portabilidad.

Por último, la lógica de la vista se puede transformar en un archivo de configuración sencillo, sin necesidad de programarla. La vista se encarga de producir las páginas que se muestran como resultado de las acciones. Esta transforma el modelo en una página Web que le permite al usuario interactuar con ella. Contiene tres partes fundamentales, el layout (es la plantilla global), el complemento de las acciones (llamado plantillas) y las páginas con sus formularios.

El layout contiene el código HTML que es común en todas las páginas, para no tener que repetirlo en todas ellas. El contenido de la plantilla se integra en el layout. El complemento de las acciones o plantillas son las encargadas de con los resultados de la acción, insertarse en el cuerpo del layout y generar finalmente la página Web resultado de la petición de un usuario. Las Páginas Clientes y Formularios son las páginas que se generan finalmente y con las que el usuario interactúa. **(Potencier, y otros, 2008)**

Para el desarrollo de las interfaces del Módulo para la gestión de metadatos geográficos de LiberMaps se utiliza EXTJS que es una librería Javascript que incorpora la tecnología AJAX para construir aplicaciones complejas con mayor eficiencia, ello influye en el diseño de la aplicación por tanto es preciso detallar la

forma en que interactúa EXTJS y Symfony. La utilización de la librería se expresa a través de un paquete que representa todos los componentes de la propia librería, el paquete ExtJs.

En cada caso de uso se utilizan un número de ficheros JS comunes a todos, encargados de configurar algunos elementos generales del funcionamiento de la aplicación. Al mismo tiempo, en cada caso de uso se utilizan otros ficheros JS específicos, los cuales contienen la configuración para las acciones de cada uno de estos en particular. El paquete de JS comunes del que hacen uso todos los CU queda representado en la figura 9.

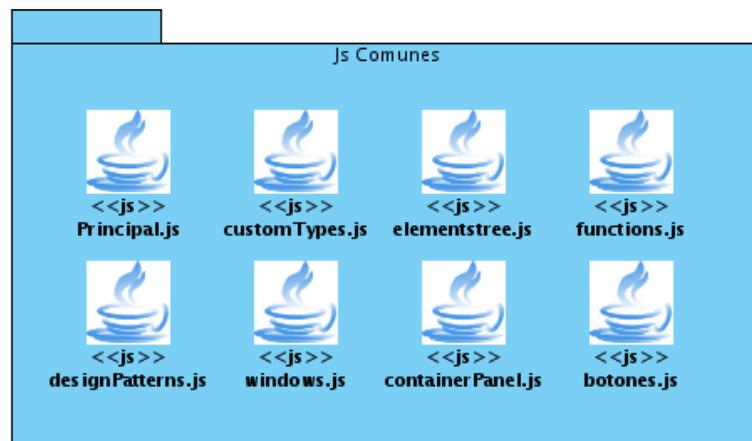


Figura 9: Paquete de JS comunes a todos los CU.

La estructura general del Módulo para la gestión de metadatos geográficos de LiberMaps puede apreciarse a través del siguiente Diagrama de Paquetes:

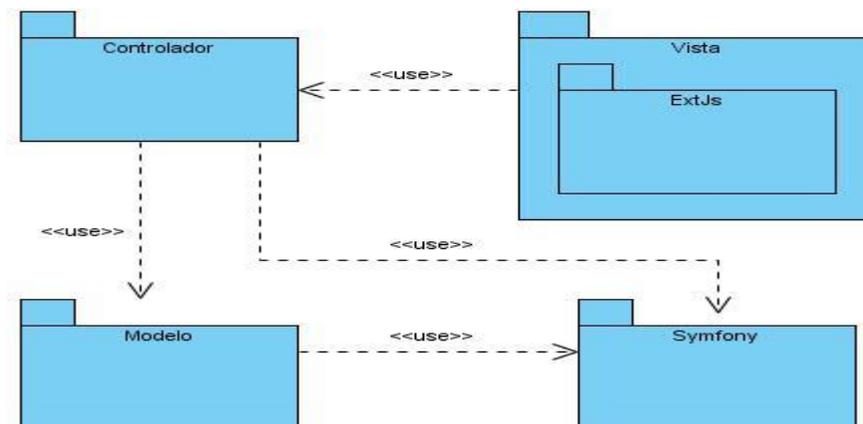


Figura 10: Estructura del Módulo para la gestión de metadatos geográficos de LiberMaps.

4.3. Patrones de Diseño.

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. No son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. **(Jacobson, y otros, 2000)** Symfony implementa un conjunto de patrones de diseño los cuales se describen en este epígrafe, así como los patrones empleados con javascript en la interfaz gráfica del sistema.

4.3.1. Patrones GRASP implementados en Symfony

Creador

En la clase Actions se encuentran todas las acciones definidas y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

Experto

Este es uno de los más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

Alta Cohesión

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Controlador

Todas las peticiones Web son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Bajo Acoplamiento

La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

4.3.2. Tipos de patrones GOF que implementa Symfony

En la categoría Creacionales:

Instancia única (Singleton):

Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a sfContext::getInstance(). En una acción, el método getContext(), es un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony.

Fábrica abstracta (Abstract Factory)

Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

En la categoría Estructurales:

Objeto compuesto (Composite)

Permite tratar objetos compuestos como si se tratase de uno simple. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera. **(Rabaix, 2008)**

4.3.3. Patrones implementados en la vista de la aplicación.

Fachada (Facade)

La fachada brinda una interfaz sencilla que hace de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. En esta aplicación sirve de intermediario entre la interfaz gráfica y los controladores de Symfony de forma tal que hace el código más legible y por ello puede reducir la dependencia de código externo, ya que la mayoría del código lo usa la fachada, permitiendo así más flexibilidad en el desarrollo del sistema.

4.4. Diagramas de Clases del Diseño.

Dentro de los artefactos que genera el modelo de diseño se encuentra el Diagrama de Clases del Diseño, en el cual se muestran un conjunto de clases, interfaces y colaboraciones, así como las relaciones entre sí. Se realiza cada diagrama por caso de uso identificado, estos diagramas son importantes pues visualizan, especifican y documentan modelos estructurales, lográndose una muestra amplia y confiable del sistema previo a su implementación. En este epígrafe se muestra el diagrama de clases del diseño del Caso de Uso Gestionar Metadato y el paquetes Modelo correspondiente a este.

4.4.1. Realización del Caso de Uso: Gestionar Metadato.

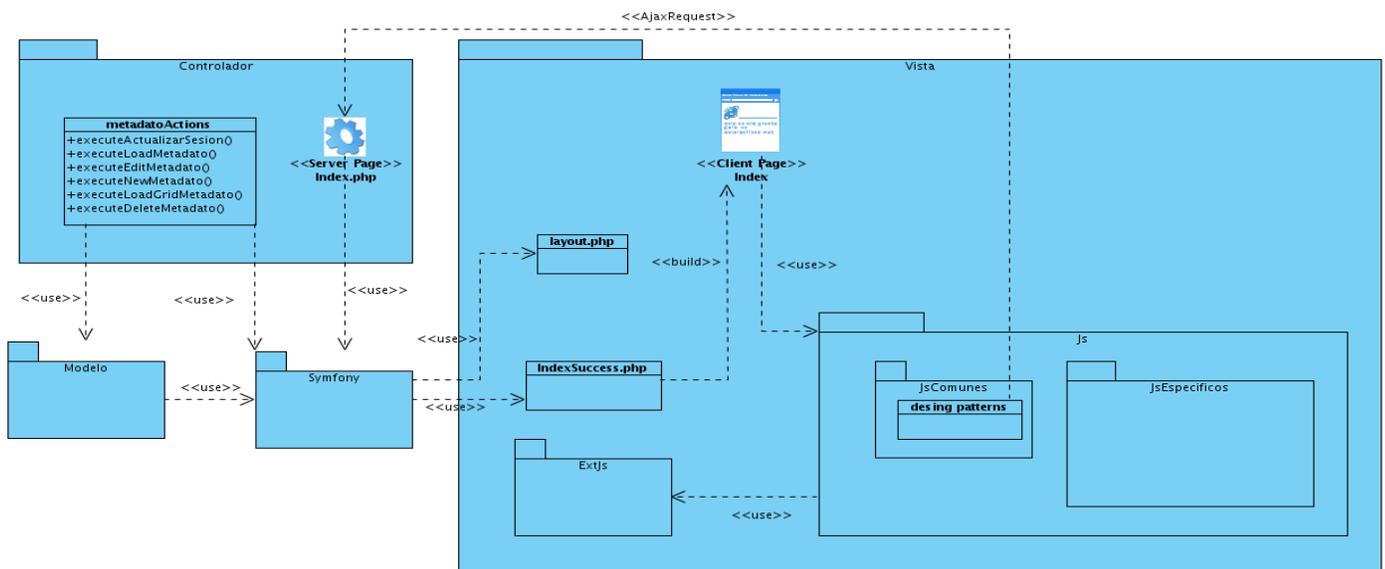


Figura 11: Diagrama de Clases del Diseño CU Gestionar Metadato.

4.4.1.1. Paquete Modelo del Caso de Uso Gestionar Metadato.

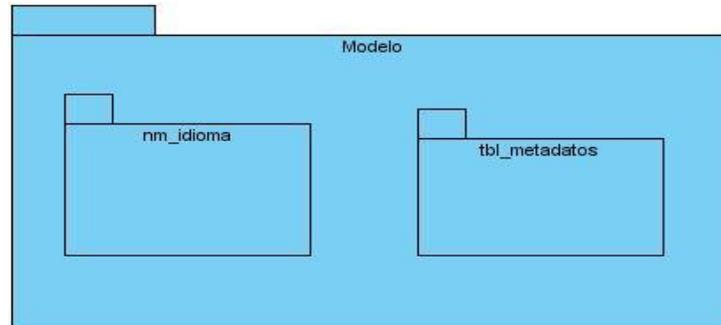


Figura 12: Paquete Modelo CU Gestionar Metadato.

4.5. Diseño de la Base de Datos.

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Para modelar el diseño de la base de datos primeramente se realiza el diagrama de clases persistentes a partir de las clases del diseño que tienen un carácter permanente y luego se lleva a cabo la realización del modelo de datos del sistema. Un Modelo de Datos se puede definir como un conjunto de conceptos, reglas y convenciones bien definidos que permiten aplicar una serie de abstracciones a fin de describir y manipular los datos de un cierto mundo real que se desea almacenar en la base de datos.

A continuación se presenta el diagrama de clases persistentes y el modelo entidad-relación correspondiente a la entidad metadato, para una mejor visualización y comprensión de los mismos.

4.5.1. Diagrama de Clases Persistentes

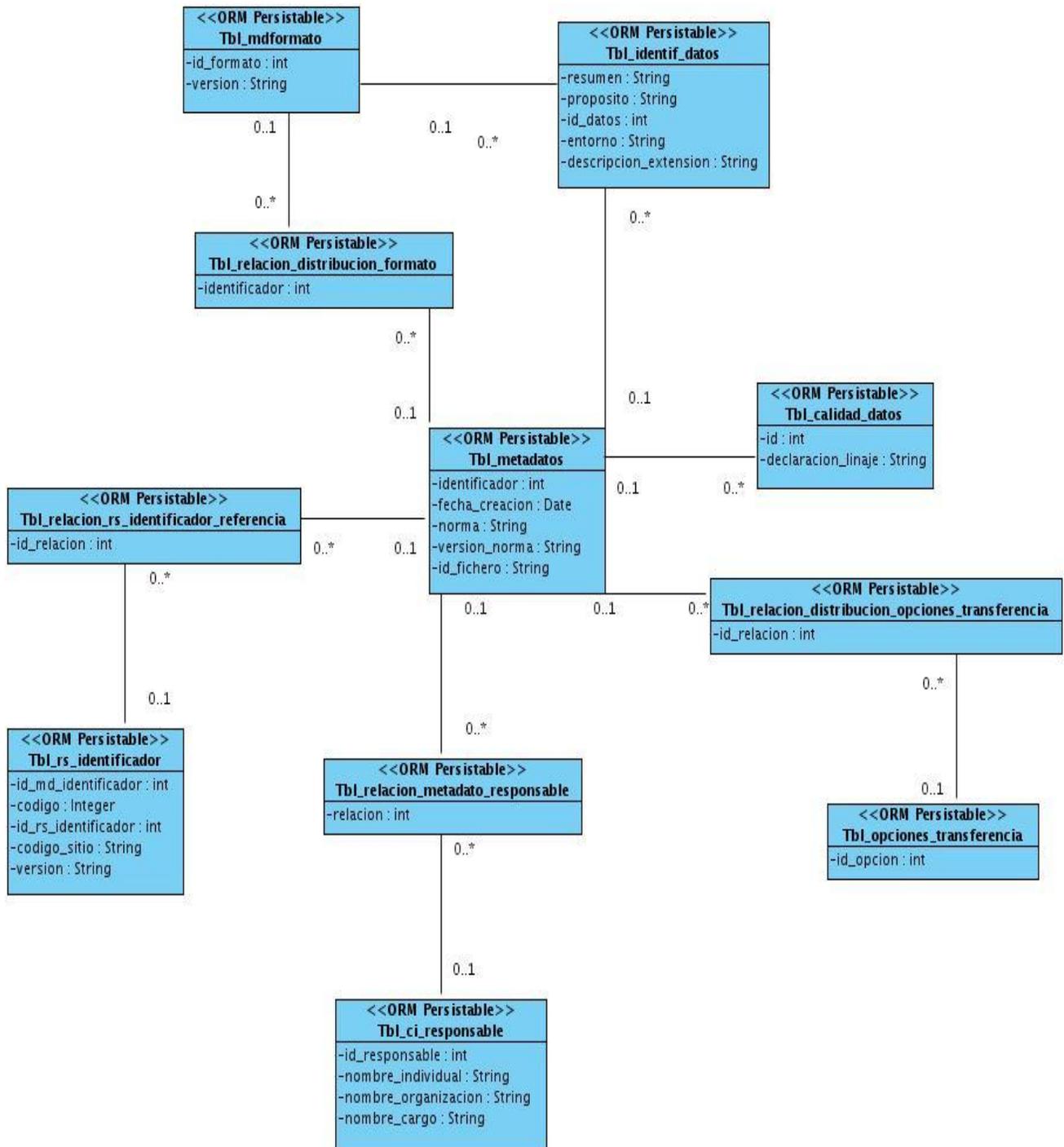


Figura 13: Diagrama de Clases Persistentes. Paquete Información del Metadato.

4.5.2. Modelo Entidad-Relación

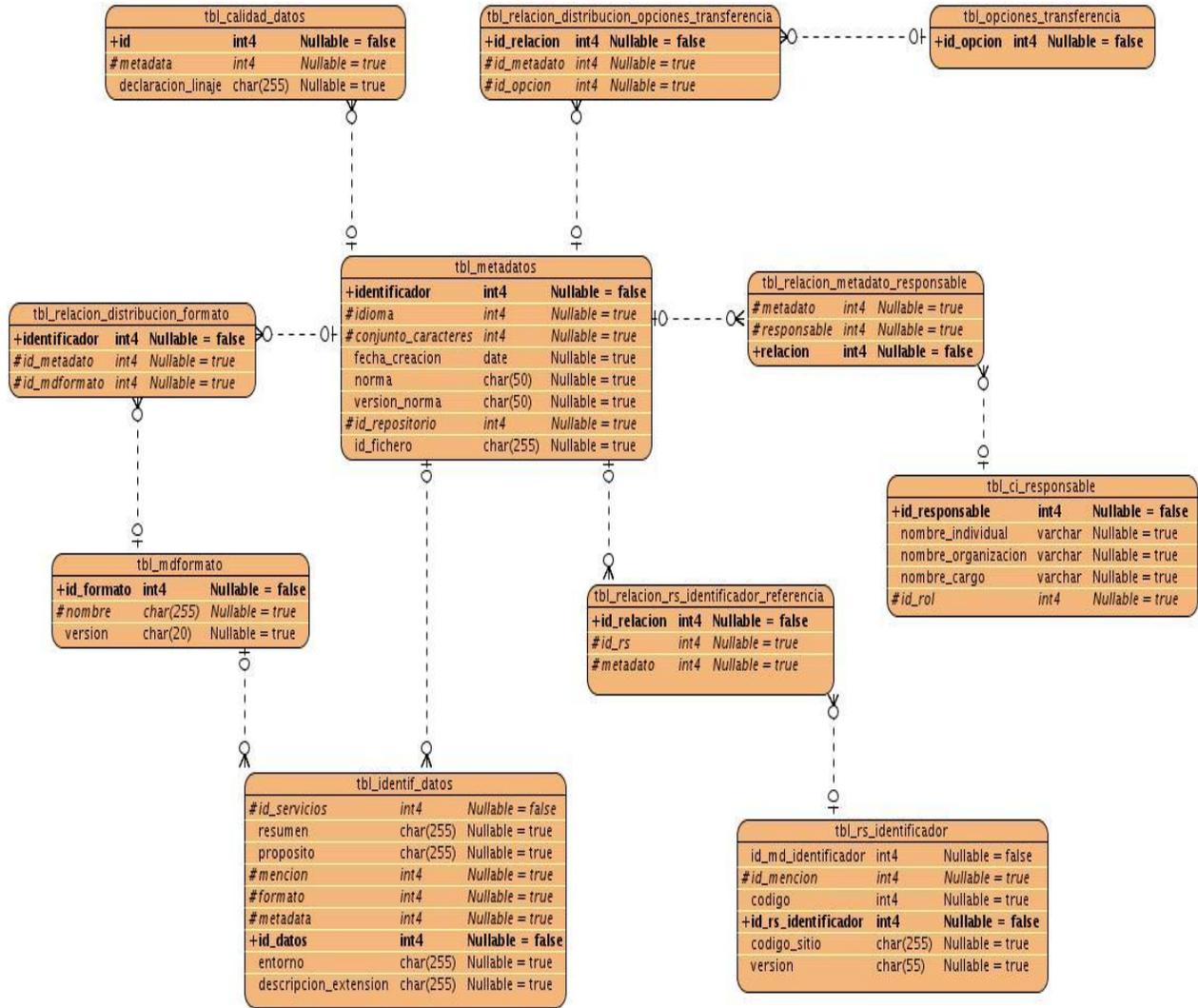


Figura 14: Modelo Entidad Relación. Paquete Metadato

4.6. Modelo de Despliegue

El Modelo de Despliegue es un modelo de objetos que representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. **(38)**

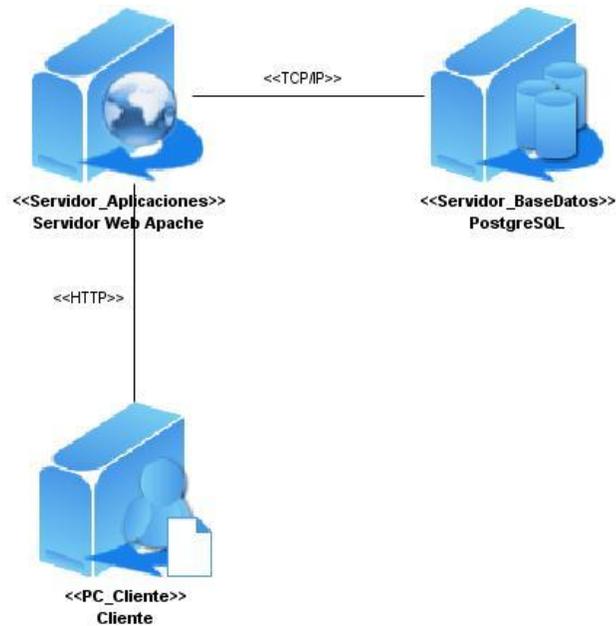


Figura 15: Diagrama de Despliegue.

4.7. Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros. **(Jacobson, y otros, 2000)** A continuación se muestra una vista general del modelo de implementación del Módulo para la gestión de metadatos geográficos de LiberMaps.

Vista General.

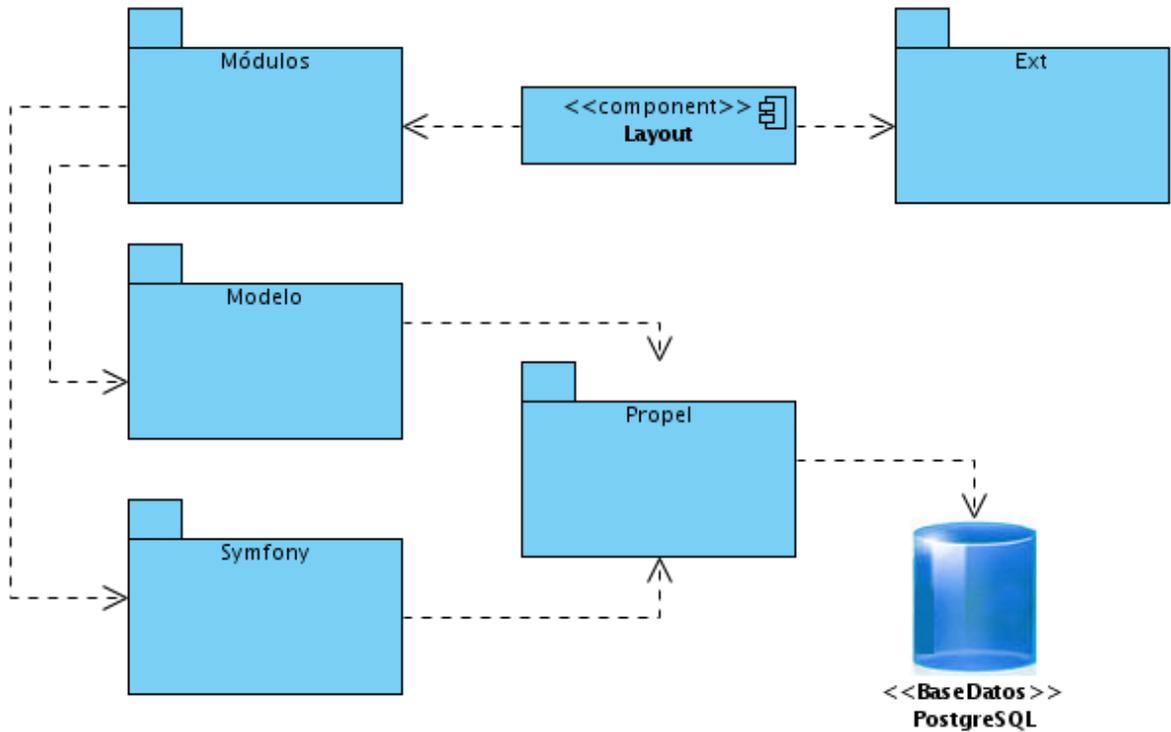


Figura 16: Diagrama de Componentes. Vista General.

Para una mayor comprensión del Diagrama de Componentes, se expone una vista detallada de los paquetes Modelo, Ext, Módulos y los componentes que contiene cada Módulo en particular.

Vista Detallada: Paquete Modelo.

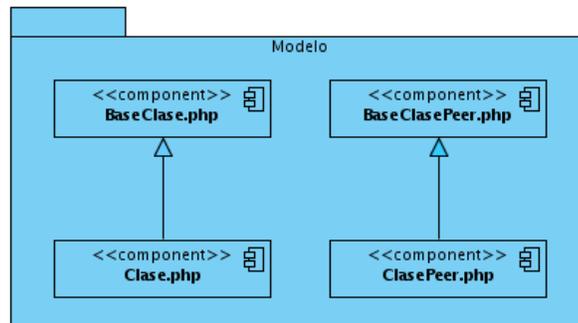


Figura 17: Diagrama de Componentes. Paquete Modelo.

Vista Detallada: Paquete Ext.

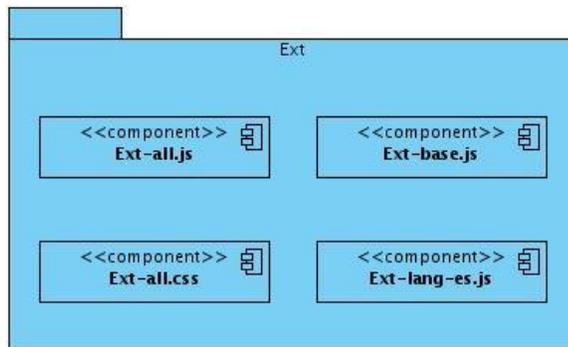


Figura 18: Diagrama de Componentes. Paquete Ext.

Vista Detallada: Paquete Módulos

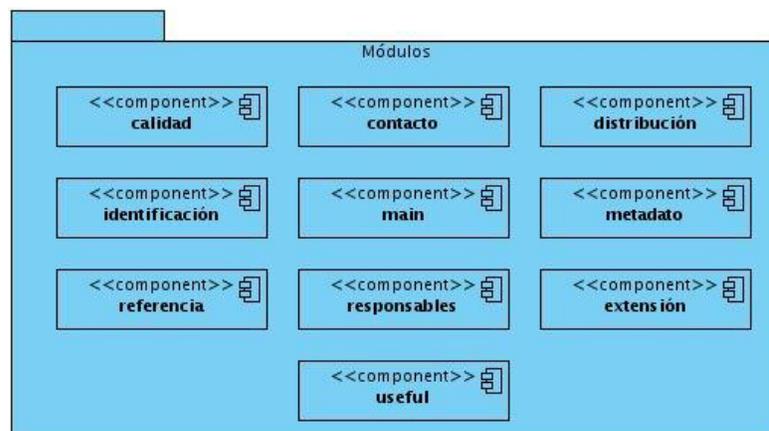


Figura 19: Diagrama de Componentes. Paquete Módulos.

Vista Detallada de cada Módulo

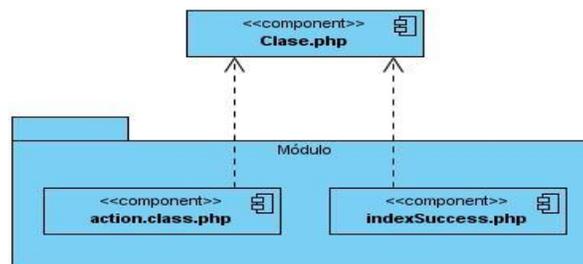


Figura 20: Diagrama de Componentes. Paquete Módulo.

4.8. Conclusiones

Durante el desarrollo de este capítulo se llevaron a cabo los flujos de trabajo: Análisis y Diseño e Implementación. Como resultado de estos flujos se generaron los artefactos pertinentes a cada uno de ellos como son el Modelo de Diseño, el cual permitió representar la estructura del Módulo para la gestión de metadatos geográficos de LiberMaps mostrando sus clases, atributos y las relaciones entre ellos. Se realizó el Modelo de Despliegue a través del cual se muestra la distribución física del sistema y sus conexiones. Mediante la realización del Modelo de Datos se logró describir los elementos que intervienen en el problema dado y la forma en que se relacionan estos elementos entre sí. Finalmente se expuso el Modelo de Implementación, donde se especificaron los distintos componentes creados para desarrollar la aplicación y la relación entre ellos, lográndose traducir el diseño en términos de componentes ejecutables. De esta forma se cuenta con la base principal en la cual apoyar la construcción del sistema.

CONCLUSIONES

El uso de la información geográfica, así como su creación y distribución, ha crecido exponencialmente, haciéndose preciso documentarla para facilitar su organización y el acceso a la misma. Los metadatos geográficos constituyen la solución a este problema.

Durante el desarrollo de esta investigación se logró cumplir con el objetivo general propuesto: “Desarrollar un módulo para el catálogo de mapas LiberMaps que permita la gestión de metadatos geográficos”. El mismo permite documentar la información geográfica y facilita una mejor organización y localización de los datos.

El hecho de que el sistema esté desarrollado con una arquitectura web, posibilita explotar los beneficios de la red, minimizando los costos de hardware en las organizaciones cubanas. No obstante, el alcance del producto no debe limitarse a territorio nacional, ya que es potencialmente exportable tanto como solución independiente o integrada a un SIG como lo es GeneSIG.

La aplicación fue desarrollada mediante herramientas multiplataforma lo que permite que pueda ser desplegada sobre entornos libres, de esta forma se garantiza un producto con mayor tiempo de vida, más reutilizable y eficiente. Además, le permite al país contar con una herramienta propia que responda a los intereses de sus organizaciones, y que esté respaldada por un equipo de desarrollo cubano que pueda dar soporte y mantenimiento a esta solución, problema muy común en la comunidad de software libre.

Teniendo en cuenta estos aspectos, y además que Cuba está inmersa en la creación de una Infraestructura de Datos espaciales, contar con una herramienta para la creación de metadatos propia es un gran reto y oportunidad, que sin dudas posee un impacto social considerable, ya que no existe otra solución cubana similar. Esta además ayuda en la informatización de las esferas socioeconómicas del país, que hoy se sustentan en software propietario para desarrollar estas tareas, y en peores casos, para suplir el trabajo manual de algunas entidades que presentan un deterioro en su productividad y desempeño, por la falta de sistemas que automaticen sus labores y apoyen la toma de decisiones, muchas veces de gran impacto en la sociedad.

RECOMENDACIONES

Luego de haber concluido el sistema propuesto y cumplido los objetivos trazados, se plantean las siguientes recomendaciones:

- ❖ Someter el sistema a pruebas de calidad de software.
- ❖ Adicionarle un componente para la publicación de metadatos geográficos basado en la herramienta Geonetwork.
- ❖ Continuar el desarrollo de la aplicación llevando a cabo la implementación de las funcionalidades que no fueran resueltas en la presente investigación y que forman parte del conjunto de elementos del núcleo de la norma ISO19115.
- ❖ Implementar la norma ISO 19115 en su totalidad así como la ISO15836: Dublin Core y especificaciones de países u organizaciones internacionales.
- ❖ Realizar la integración del sistema con el catálogo de mapas LiberMaps.

REFERENCIAS BIBLIOGRÁFICAS

1. Almada, Federico. 2008. Techtear.com Blog Magazine de tecnología. Zend Studio for Eclipse, desarrollo profesional en PHP. [En línea] 22 de enero de 2008. [Citado el: 20 de diciembre de 2009.] <http://www.techtear.com/2008/01/22/zend-studio-for-eclipse-desarrollo-profesional-en-php/>.
2. Angel, Miguel Alvarez. 2006. DesarrolloWeb.com. Una introducción meramente conceptual al potente lenguaje de script del lado del cliente. [En línea] 2006. [Citado el: 23 de diciembre de 2009.] <http://www.desarrolloweb.com/articulos/25.php>.
3. Apache. 2007. Servidor web Apache. [En línea] 2007. [Citado el: 3 de diciembre de 2009.] <http://www.apache.org>.
4. Bohorquez, Hanller Cueto y Valenzuela, Johan Ferney. 2005. Telecentro regional de Tecnologías Espaciales. Fundamentos de Infraestructura de Datos Espaciales -01(FIDE -01). [En línea] 2005. [Citado el: 17 de noviembre de 2009.] <http://geoservice.igac.gov.co/moodle/course/info.php?id=23>.
5. CHACÓN, JULIO CÉSAR RUEDA. 2006. APLICACIÓN DE LA METODOLOGÍA RUP PARA EL DESARROLLO RÁPIDO DE APLICACIONES BASADO EN EL . Guatemala : Universidad de San Carlos de Guatemala. Facultad de Ingeniería. Escuela de Ingeniería en Ciencias , 2006.
6. Cisneros Díaz, Luís R. y Duffatt Candeaux, Rafael. Julio de 1994. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG: DEFINICIÓN, CARACTERÍSTICAS, ESTADO ACTUAL Y TENDENCIAS DE DESARROLLO. s.l. : Mapping Interactivo. Revista Internacional de Ciencias de la Tierra., Julio de 1994.
7. COMISIÓN DE LAS COMUNIDADES EUROPEAS. 2008. REGLAMENTO (CE) No 1205/2008 DE LA COMISIÓN de 3 de diciembre de 2008 por el que se ejecuta la Directiva. 2008.
8. Eguíluz, Javier Pérez. 2006. LibrosWeb.es. Introducción a AJAX. [En línea] 2006. [Citado el: 22 de diciembre de 2009.] <http://librosweb.es/ajax/index.html>.

9. Enciclopedia Libre Universal en Español. Enciclopedia Libre Universal en Español. Lenguaje de programación. [En línea] [Citado el: 18 de diciembre de 2009.] http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n.
10. Fernández, Sagrario Peralta. 2008. El servidor Web. Arquitectura y funcionamiento. [En línea] 2008. [Citado el: 3 de diciembre de 2009.] <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=366>.
11. Franco, Rodolfo. 2009. Universidad Distrital Francisco Jose de Caldas. [En línea] 11 de agosto de 2009. [Citado el: 17 de noviembre de 2009.] <http://www.udistrital.edu.co/comunidad/profesores/rfranco/metadatos.htm>.
12. González, Rubén Blanco y Pérez, Sergio Tobalina. 2005. Introducción a Rational Rose. s.l. : Facultad de Informatica de Barcelona. Departamento de Lenguajes y Sistemas informáticos.Laboratorio de Ingeniería de Software., 2005.
13. Gracia, Joaquin. 2004. WebEstilo Usabilidad, programación y mucho mas. Vista general de ASP.NET. [En línea] 3 de Abril de 2004. [Citado el: 20 de diciembre de 2009.] <http://www.webestilo.com/aspnet/aspnet01.phtml>.
14. IDEE. Portal IDEE. Infraestructura de Datos Espaciales de España. [En línea] [Citado el: 20 de noviembre de 2009.] http://www.idee.es/show.do?to=pideep_que_es_INSPIRE.ES.
15. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. Madrid : Series Editors, 2000.
16. Maganto, Alejandra Sánchez y Ballari, Daniela. 2008. Mapping Interactivo. Revista Internacional de Ciencias de la Tierra. NORMAS SOBRE METADATOS (ISO 19115, ISO 19115-2, ISO 19139, ISO 15836). [En línea] Enero de 2008. [Citado el: 20 de noviembre de 2009.] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1455.

17. Manual de sistemas de información geográfica y cartografía digital. PUBLICACIÓN, DE LAS NACIONES UNIDAS. 2000. 79, Nueva York : Departamento de Asuntos Económicos y Sociales. División de Estadística, 2000, Vol. Serie F.
18. Martinez, Rafael. 2009. PostgreSQL-es.org Portal en español sobre PostgreSQL. Sobre PostgreSQL. [En línea] 22 de marzo de 2009. [Citado el: 19 de diciembre de 2009.] http://www.postgresql-es.org/sobre_postgresql.
19. MySQL-Hispano.org. 2005. WebEstilo Usabilidad, programación y mucho más. Introducción a MySQL. [En línea] 9 de Junio de 2005. [Citado el: 22 de diciembre de 2009.] <http://www.webestilo.com/mysql/intro.phtml>.
20. Nogueira, Helton Uchoa, Nunes, Jorge Luís y Brito, Silva. 2005. Cartografia.cl. Análisis del módulo PostGIS (OpenGIS). [En línea] 2005. [Citado el: 19 de diciembre de 2009.] http://www.cartografia.cl/index.php?option=com_content&task=view&id=78&Itemid=9.
21. Ochoa, Jesus Alberto. 2005. VII SEMESTRE INGENIERIA DE SISTEMAS. INGENIERIA DE SOFTWARE II. [En línea] 2005. [Citado el: 18 de diciembre de 2009.] <http://members.fortunecity.com/software1/herramie.htm>.
22. Orallo, Enrique Hernández. 2007. El Lenguaje Unificado de Modelado (UML). 2007.
23. Popkin Software and Systems. 2008. Modelado de Sistemas con UML. 2008.
24. Potencier, Fabien y Zaninotto, François. 2008. Symfony la guía definitiva. 2008.
25. Programa Geodato. 2007. Procedimiento para la Documentación del Metadato. s.l. s.l. : Geominera Oriente, 2007.

26. Rabaix, Thomas. 2008. Patrones de diseño que implementa symfony. [En línea] 2008. [Citado el: 15 de Enero de 2010.] <http://www.mail-archive.com/symfony-users@googlegroups.com/msg07161.html>.
27. Ramos. 2007. Cassandra Clinic. Sistema de gestión de la información para clínicas imagenológicas. La Habana : Universidad de las Ciencias Informáticas, 2007.
28. Real Academia Española. DICCIONARIO DE LA LENGUA ESPAÑOLA. Vigésima segunda edición. . [En línea] [Citado el: 17 de noviembre de 2009.] <http://buscon.rae.es/drae/>.
29. Rodas, Raul Hinostraza. 2007. www.linuxcentro.net. Usuarios y Grupos en Linux. [En línea] 2007. [Citado el: 21 de diciembre de 2009.]
30. <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
31. Secretaría del CTN50. AENOR. FESABID. 2008. Información y documentación - Procesos de gestión de documentos - Metadatos para la gestión de documentos. Parte 1: Principios. ISO 23081-1:2006. 2. Madrid : s.n., 2008.
32. Sencha Solutions. 2008. ExtJs. Documentación Ext JS JavaScript Library. [En línea] 2008. [Citado el: 21 de diciembre de 2009.] <http://extjs.com/deploy/dev/docs>.
33. Sicilia, Miguel Angel. 2008. Conexions. Funciones de los Sistemas Gestores de Bases de Datos. [En línea] 23 de septiembre de 2008. [Citado el: 20 de diciembre de 2009.] <http://cnx.org/content/m17543/latest/>.
34. Thompson, Ivan. 2008. Portal de Mercadotecnia. [En línea] Octubre de 2008. [Citado el: 15 de Noviembre de 2009.] <http://www.promonegocios.net/mercadotecnia/definicion-informacion.html>.
35. Torres Gastelú, Dr. Carlos A. 2007. Universidad Veracruzana Facultad de Administración. Sistemas Computacionales Administrativos E.E Base de Datos. Apache e Internet Information Services. [En línea] 2007. [Citado el: 20 de diciembre de 2009.] <http://www.slideshare.net/equipo9/apache-e-iis>.

36. Valdés, Damián Pérez. 2007. Maestros del Web. Los Frameworks de PHP agilizan tu trabajo. [En línea] 31 de julio de 2007. [Citado el: 20 de diciembre de 2009.] <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
37. Visual Paradigm International. 2010. Visual Paradigm. Visual Paradigm for UML - UML tool for software application development. [En línea] 11 de Enero de 2010. [Citado el: 19 de diciembre de 2009.] <http://www.visual-paradigm.com/product/vpum/>.
38. Wheeler, Veronika. 2008. Linux en Español. Entorno de desarrollo integrado. [En línea] 26 de marzo de 2008. [Citado el: 20 de diciembre de 2009.] <http://www.linuxespanol.com/viewtopic.php?p=96480&sid=c1e2f61c2c9f25c5ca0e492a6c87c62e>.
39. Zayas, Dr. Cs. Carlos Alvarez de. 1995. METODOLOGIA DE LA INVESTIGACION CIENTIFICA. Santiago de Cuba : CENTRO DE ESTUDIOS DE EDUCACION SUPERIOR "MANUEL F. GRAN", 1995.

BIBLIOGRAFÍA

1. Bohorquez, Hanller Cueto y Valenzuela, Johan Ferney. Telecentro regional de Tecnologías Espaciales. Fundamentos de Infraestructura de Datos Espaciales -01(FIDE -01). [En línea] [Citado el: 17 de noviembre de 2009.] <http://geoservice.igac.gov.co/moodle/course/info.php?id=23>.
2. CE. 2008. COMISIÓN DE LAS COMUNIDADES EUROPEAS. REGLAMENTO (CE) No 1205/2008 DE LA COMISIÓN de 3 de diciembre de 2008 por el que se ejecuta la Directiva. 2008.
3. DICCIONARIO DE LA LENGUA ESPAÑOLA. Vigésima segunda edición. [En línea] [Citado el: 17 de noviembre de 2009.] <http://buscon.rae.es/drael/>.
4. Franco, Rodolfo. 2009. Universidad Distrital Francisco Jose de Caldas. [En línea] 11 de agosto de 2009. [Citado el: 17 de noviembre de 2009.] <http://www.udistrital.edu.co/comunidad/profesores/rfranco/metadatos.htm>.
5. IDEE. Portal IDEE. Infraestructura de Datos Espaciales de España. [En línea] [Citado el: 20 de noviembre de 2009.] http://www.idee.es/show.do?to=pideep_que_es_INSPIRE.ES.
6. Información y documentación - Procesos de gestión de documentos - Metadatos para la gestión de documentos. Parte 1: Principios. ISO 23081-1:2006. 2. Secretaría del CTN50. AENOR. FESABID. 2008. Madrid : s.n., 2008, Vol. 31.
7. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. Madrid : Series Editors, 2000.
8. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG: DEFINICIÓN, CARACTERÍSTICAS, ESTADO ACTUAL Y TENDENCIAS DE DESARROLLO. Cisneros, Luís R. Díaz y Duffatt, Rafael Candeaux. Julio de 1994. s.l. : Mapping Interactivo. Revista Internacional de Ciencias de la Tierra., Julio de 1994, Vols. ISSN: 1.131-9.100, s.l.

9. Maganto, Alejandra Sánchez y Ballari, Daniela. 2008. Mapping Interactivo. Revista Internacional de Ciencias de la Tierra. NORMAS SOBRE METADATOS (ISO 19115, ISO 19115-2, ISO 19139, ISO 15836). [En línea] Enero de 2008. [Citado el: 20 de noviembre de 2009.] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1455.
10. Manual de sistemas de información geográfica y cartografía digital. PUBLICACIÓN, DE LAS NACIONES UNIDAS. 2000. 79, Nueva York : Departamento de Asuntos Económicos y Sociales. División de Estadística, 2000, Vol. Serie F.
11. Potencier, Fabien y Zaninotto, François. 2008. Symfony la guía definitiva. 2008.
12. Programa Geodato. 2007. Procedimiento para la Documentación del Metadato. s.l. s.l. : Geominera Oriente, 2007.
13. Sencha Solutions. 2008. ExtJs. Documentación Ext JS JavaScript Library. [En línea] 2008. [Citado el: 21 de diciembre de 2009.] <http://extjs.com/deploy/dev/docs>.
14. Thompson, Ivan. 2008. Portal de Mercadotecnia. [En línea] Octubre de 2008. [Citado el: 15 de Noviembre de 2009.] <http://www.promonegocios.net/mercadotecnia/definicion-informacion.html>.
15. Zayas, Dr. Cs. Carlos Alvarez de. 1995. METODOLOGIA DE LA INVESTIGACION CIENTIFICA. Santiago de Cuba : CENTRO DE ESTUDIOS DE EDUCACION SUPERIOR "MANUEL F. GRAN", 1995.

GLOSARIO DE TÉRMINOS

Término	Definición
ACID	Atomicidad, Consistencia, Aislamiento y Durabilidad. Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción. Entonces si un sistema de gestión de bases de datos es <i>ACID</i> , esto quiere decir que el mismo cuenta con las funcionalidades necesarias para que sus transacciones tengan las características ACID.
CGI	Pasarela de Interface Común, o en sus siglas en inglés: Common Gateway Interface.
CU	Caso de Uso.
Datos ráster	Tipo de imagen digital representada en mallas, es decir, a través de matrices.
DCUS	Diagrama de Casos de Uso del Sistema.
GeneSIG	Plataforma que sirve de apoyo al desarrollo de SIG en entornos web con tecnologías libres.
HTML	Lenguaje de Marcado de Hipertexto, o en sus siglas en inglés: HyperText Markup Language.
HTTP	Protocolo de Transferencia de Hipertexto, o en sus siglas en inglés: Hypertext Transfer Protocol.
IDE	Infraestructura de Datos Espaciales.
ISO	Organización Internacional de Normalización.
Imágenes de satélite	Es un barrido de sensores de distinta longitud de onda que generan una matriz de valores, es decir, es una captura del espectro visible sobre un plano.
INSPIRE	Infraestructura de información espacial en Europa, o en sus siglas en inglés:

	Infraestructure for Spatial Information in Europe.
Modelos digitales	Estructuras numéricas de datos que representa la distribución espacial de una variable cuantitativa y continua, como puede ser la temperatura, la cota o la presión atmosférica.
MVC	Modelo-Vista-Controlador. Patrón arquitectónico de diseño.
NEM	Núcleo Español de Metadatos.
Orto-fotografías	Imagen a la que se le han corregido sus deformaciones geométricas.
ONRM	Oficina Nacional de Recursos Minerales.
ORM	Mapeo Objeto-Relacional, o en sus siglas en inglés: Object-Relational Mapping.
RUP	Proceso Unificado de Rational, o en sus siglas en inglés: Rational Unified Process.
RF	Requisito Funcional.
RNF	Requisito no Funcional.
SIG	Sistema de Información Geográfica.
SOA	Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture).
URL	Localizador de Recurso Uniforme (en inglés <i>Uniform Resource Locator</i>), la dirección global de documentos y de otros recursos en la World Wide Web.
UML	Lenguaje Unificado de Modelado.
WISE	Sistema de Información Hidráulica para Europa, o en sus siglas en inglés: Water Information System for Europe.
XP	Programación Extrema, o en sus siglas en inglés: Extreme Programming.
XML	Lenguaje extensible de Marcado, o en sus siglas en inglés: Extensible Markup Language.

Tabla 3: Glosario de Términos.