



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 9

# Subsistema para la consulta de metadatos geológicos en La Oficina Nacional de Recursos Minerales

---

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

**Autores: Lisbeth O. López Verdecie  
José L. Muñoz Suárez**

**Tutor: Lic. Osmel de la Cruz Amador**

**Co-Tutora: Ing. Eylin Hernández Luque**

*Julio de 2008*

*NUESTRA RECOMPENSA SE ENCUENTRA EN EL ESFUERZO Y NO EN EL RESULTADO.  
UN ESFUERZO TOTAL ES UNA VICTORIA COMPLETA.*

*MAHATMA GANDHI*

*A mi madre, a mis abuelos.*  
*Jose*

*A mis padres, a mi hermana.*  
*Lisbeth*

# ***Agradecimientos***

A la Revolución por darnos esta oportunidad...

A todos aquellos que contribuyeron de una u otra forma en la realización de este trabajo.

*De Jose Luis*

A mi familia y especialmente a mi madre por educarme y darme tanto amor.

A mis abuelos por malcriarme y quererme tanto.

A mi padre que a pesar de todo también influyó en mi educación.

A mis vecinos sobre todo a tía Celaida por ser parte también de mi familia.

A mis amigos, los del barrio, los del pre y los de la UCI,  
por compartir los buenos y los malos momentos.

A mis profesores, en especial a Luis Enrique, Valle y Gago, por servirme de guías.

A mi compañera de tesis Lisbeth por compartir estos estresantes momentos.

*De Lisbeth*

A mi papá, por enseñarme a crecer ante las dificultades, a mi mamá, por creer en mí siempre

A mi hermana, por “soportarme” en cada momento, a los tres, por su confianza y amor  
incondicional.

A mis tíos Bertha y Navarro, así como mis primos Lioniv y Andrei por haberme apoyado y querido  
como una hija y hermana durante estos cinco años.

A mi abuela Olga y mis tíos Olinda, Nano e Isa, por su cariño y comprensión.

A mi abuelo Pedrito, que ya no está, pero con su sabiduría y amor me hizo ser mejor.

A mis demás abuelos y tío que no están conmigo pero siempre me apoyaron.

A mi familia más cercana y a América, mi madrina, por su constante apoyo.

A mis amistades de siempre, Amalita, Elsy, Daily, Osmany, Victor, Ronny, Alexey, Yeranddy y a las  
nuevas que hice en la UCI.

A mi compañero de tesis Jose, porque además de compartir estos estresantes momentos  
contribuyó a mi formación como ingeniera.

A todos muchas gracias.

## **Declaración de Autoría**

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_ del año \_\_\_\_.

\_\_\_\_\_

Lisbeth Olinda López Verdecie

\_\_\_\_\_

José Luis Muñoz Suárez

\_\_\_\_\_

Osmel de la cruz Amador

## Datos de contacto

**Síntesis del tutor:** Lic. Osmel de la Cruz Amador

Profesión: Graduado en Ciencias de la Computación.

Categoría docente: Adiestrado

Años de graduado: 2

Responsabilidad: Jefe del proyecto Metadatos del Polo Geoinformática.

**Síntesis de la Co – Tutora:** Ing. Eyllin Hernández Luque

Profesión: Ingeniero en Ciencias Informáticas

Categoría docente: Adiestrado

Años de graduado: 1

Responsabilidad: Líder del Polo Geoinformática de la Facultad 9

**Síntesis del Consultante:** Ing. Héctor Rodríguez

Profesión: Dr. En Ciencias Geológicas

Categoría docente: Profesor Titular

Años de graduada: 30

Responsabilidad: Grupo de Desarrollo y Geomática. Oficina Nacional de Recursos Minerales.

**Síntesis del Consultante:** Ing. César Rosales

Profesión: Ingeniero Geofísico

Categoría docente: No tiene

Años de graduada: 31

Responsabilidad: Jefe del Grupo de Servicios Informáticos de la Empresa Geominera Oriente.

## Opinión del Tutor

La era de la información, como se ha dado a llamar el complejo fenómeno que experimenta la presente generación y que es producto del acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) tiene en el escenario nacional la tarea de informatizar el patrimonio económico y social del que forma parte el conocimiento geológico propio de la Oficina Nacional de Recursos Minerales (ONRM), perteneciente al Ministerio de la Industria Básica (MINBAS).

La Universidad de las Ciencias Informáticas en su misión de formar profesionales sobre la práctica y contribuir con el desarrollo económico del país ha asumido la realización de esta tarea, en el caso particular del presente trabajo se constituye la solución del diseño de un sistema capaz de gestionar las consultas a los metadatos geológicos.

Los diplomantes Lisbeth O. López Verdecie y José L. Muñoz Suárez, han demostrado compromiso y esmero en la realización de la investigación científica. Su dinamismo y laboriosidad han sido un factor fundamental en el cumplimiento exitoso del objetivo trazado en dicha investigación. Los estudiantes mostraron independencia absoluta, originalidad y autonomía en cada fase de la tarea a realizar, logrando resultados satisfactorios y buena aceptación de la información y la modelación del trabajo.

El documento realizado está debidamente estructurado, posee un enfoque científico con criterio propio por parte de los autores, buena organización y coherencia, se emplean los estilos de redacción apropiados y se evidencia el cumplimiento del objetivo propuesto.

Por todo lo anteriormente expresado, considero que los estudiantes están aptos para ejercer como Ingeniero en Ciencias Informáticas y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

\_\_\_\_\_  
Tutor: Lic. Osmel de la Cruz Amador

Junio del 2008

Fecha

## Resumen

En la actualidad, con el objetivo de materializar la idea de la informatización de la sociedad en Cuba, la Universidad de las Ciencias Informáticas lleva a cabo una serie de proyectos que contribuyen al desarrollo de las distintas esferas de la población. Uno de ellos lo constituye el Programa Nacional para la Informatización del Conocimiento Geológico, que se realiza en conjunto con la Oficina Nacional de Recursos Minerales (ONRM), con el objetivo de digitalizar toda la información geológica que se maneja en esta entidad y desarrollar aplicaciones que sean capaces de gestionarla. Esta información se encuentra almacenada en distintas bases de datos diseminadas por todo el territorio nacional y en diferentes formatos, provocando que se dificulte y sea ineficiente su consulta y recuperación.

Es por ello que, con el objetivo de documentar fácilmente todo este volumen de información se hace uso de los metadatos y se propone en la presente investigación el diseño de una aplicación web que permita la consulta de los metadatos geológicos en la ONRM.

Por tanto, en este trabajo de diploma se presentan los resultados obtenidos del estudio realizado sobre el proceso de recuperación de información en la ONRM y el uso de metadatos para la documentación de la misma. Finalmente se muestra el diseño de la propuesta del sistema, y se recomienda concluir con el proceso de desarrollo del mismo realizando su implementación.

Palabras Claves:

- Metadato
- Metadato geológico
- Recuperación de información
- Aplicación web



<b>Introducción .....</b>	<b>1</b>
<b>Capítulo 1 Fundamentación Teórica .....</b>	<b>4</b>
1.1 Introducción .....	4
1.2 Conceptos asociados al dominio del problema .....	4
1.2.1 Metadatos. Metadatos Geográficos.....	5
1.2.2 Recuperación de información.....	7
1.3 La Oficina Nacional de Recursos Minerales y el proceso de recuperación de información.....	8
1.3.1 Importancia del uso de Metadatos de Información Geográfica. [10] .....	10
1.3.2 Estándares para la documentación del metadato[16].....	12
1.4 Análisis de otras soluciones existentes .....	13
<b>Capítulo 2 Tendencias y tecnologías actuales a desarrollar .....</b>	<b>16</b>
2.1 Introducción .....	16
2.2 Aplicaciones Web .....	16
2.2.1 Características de las aplicaciones Web .....	16
2.3 Arquitectura de software .....	17
2.3.1 Arquitectura de una Aplicación Web: Modelo Cliente/Servidor.....	18
2.3.2 Modelo – Vista – Controlador (MVC) .....	20
2.4 Los diferentes lenguajes de programación para la Web .....	21
2.4.1 Lenguaje de programación del lado del servidor: PHP .....	22
2.4.2 Lenguaje de programación del lado del cliente .....	23
2.4.3 Fundamentación del lenguaje de programación a utilizar .....	23
2.5 AJAX: Nueva técnica para el desarrollo de aplicaciones Web .....	24
2.6 ¿Qué es un <i>framework</i> ? .....	26
2.6.1 <i>Framework</i> para PHP: Symfony .....	27
2.6.2 ¿Por qué Symfony? .....	29
2.7 Servidores web.....	29
2.7.1 Apache .....	29
2.8 ¿Qué es un Sistema Gestor de Bases de Datos?.....	30
2.8.1 PostgreSQL como Sistema Gestor de Base de Datos .....	30
2.8.2 Fundamentación del Sistema Gestor de Base de Datos a utilizar.....	31
2.9 El Proceso Unificado de Desarrollo de Software como metodología de desarrollo. ....	32
2.9.1 ¿Por qué RUP? .....	35
2.10 El Lenguaje Unificado de Modelado como soporte de la modelación de la solución propuesta. ....	36
2.11.1 Visual Paradigm.....	36
2.11.2 Eclipse.....	37
2.12 Conclusiones .....	38
<b>Capítulo 3 Presentación de la solución propuesta .....</b>	<b>39</b>
3.1 Introducción .....	39
3.2 Modelo del Negocio propuesto .....	39
3.2.1 Actores y Trabajadores del Negocio.....	39

3.2.2 Procesos del Negocio .....	40
3.2.3 Diagrama de Casos de Uso del Negocio.....	40
3.2.4 Descripción textual de los Casos de Uso del Negocio .....	40
3.2.5 Modelo de Objetos del Negocio.....	42
El Modelo de Objetos del Negocio muestra .....	42
3.3 Requerimientos Funcionales/No Funcionales del sistema .....	43
3.3.1 Requerimientos Funcionales .....	43
3.3.2 Requerimientos No Funcionales .....	43
3.4 Descripción del Sistema propuesto.....	45
3.4.1 Descripción de los actores.....	46
3.4.2 Casos de Usos del Sistema .....	46
3.4.2.1 Diagrama de casos de uso del sistema .....	46
3.4.2.2 Descripción de los casos de uso del sistema .....	47
3.5 Conclusiones.....	52
<b>Capítulo 4 Construcción de la solución propuesta .....</b>	<b>53</b>
4.1 Introducción .....	53
4.2 Análisis.....	53
4.2.2 Diagramas de interacción.....	53
4.3 Diseño .....	54
4.3.1 Diagramas de clases del diseño.....	54
4.4 Diseño de la Base de Datos .....	59
4.4.1 Diagrama de Clases Persistentes .....	60
4.4.2 Modelo Entidad – Relación .....	62
4.5 Modelo de despliegue .....	63
4.6 Modelo de implementación .....	64
4.7 Conclusiones.....	67
<b>Conclusiones .....</b>	<b>68</b>
<b>Recomendaciones.....</b>	<b>69</b>
<b>Referencias Bibliográficas .....</b>	<b>70</b>
<b>Glosario de Términos.....</b>	<b>73</b>
<b>Anexos.....</b>	<b>76</b>
ANEXO I. DIAGRAMAS DE CLASES DEL ANÁLISIS .....	76
ANEXO II. DIAGRAMAS DE COLABORACIÓN .....	78
ANEXO III. ENTREVISTA A ESPECIALISTAS.....	82

## Figuras

Figura 1: Arquitectura clásica cliente/servidor .....	19
Figura 2: Ciclo de vida MVC .....	20
Figura 3: Tecnologías agrupadas bajo el concepto de AJAX. ....	25
Figura 4: Comparación entre AJAX y el modelo tradicional de aplicación web. ....	25
Figura 5: RUP en dos dimensiones. ....	34
Figura 6: Diagrama de Caso de Uso del Negocio. ....	40
Figura 7: Diagrama de actividades del negocio.....	42
Figura 8: Modelo de Objeto.....	42
Figura 9: Diagrama de Casos de Uso del Sistema .....	46
Figura 10: Flujo de trabajo de Symfony. ....	55
Figura 11: DCD. CU: Consultar Metadatos por Palabras Claves.....	56
Figura 12: DCD. CU: Mostrar el Metadatos Completo. ....	56
Figura 13: DCD. CU: Salvar Metadatos. ....	57
Figura 14: DCD. CU: Mostrar el Núcleo del Metadato. ....	57
Figura 15: DCD. CU: Consultar Metadatos por Categorías. Sección: Identificación de los Datos .....	58
Figura 16: DCD. CU: Consultar Metadatos por Categorías. Sección: Calidad de los Datos. ....	58
Figura 17: DCD. CU: Consultar Metadatos por Categorías. Sección: Información del Contenido. ....	59
Figura 18: DCD. CU: Consultar Metadatos por Categorías. Sección: Información del Metadato. ....	59
Figura 19: Modelo de Datos genérico en Symfony .....	60
Figura 20: Diagrama de Clases Persistentes. ....	61
Figura 22: Modelo Entidad - Relación.....	63
Figura 23: Modelo de Despliegue.....	64
Figura 24: Diagrama de Implementación. CU: Consultar Metadatos por Palabras Claves. ....	65
Figura 25: Diagrama de Implementación. CU: Mostrar el Metadato Completo. ....	65
Figura 26: Diagrama de Implementación. CU: Salvar el Metadato. ....	66
Figura 27: Diagrama de Implementación. CU: Mostrar el Núcleo del Metadato. ....	66
Figura 28: Diagrama de Implementación. CU: Consultar Metadato por Categoría. ....	67
Figura 29: Diagramas de Clases del Análisis. CU: Consultar Metadato por Palabras Claves. ....	76
Figura 30: Diagramas de Clases del Análisis. CU: Consultar Metadatos por Categorías.....	76
Figura 31: Diagramas de Clases del Análisis. CU: Mostrar Núcleo del Metadato. ....	77
Figura 32: Diagramas de Clases del Análisis. CU: Mostrar Metadato Completo .....	77
Figura 33: Diagramas de Clases del Análisis. CU: Salvar Metadato. ....	77

Figura 34: Diagrama de colaboración. CU: Mostrar el Metadato Completo .....	78
Figura 35: Diagrama de Colaboración. CU: Mostrar el Núcleo del Metadato. ....	78
Figura 36: Diagrama Colaboración. CU: Salvar Metadato.....	79
Figura 37: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Calidad de Datos. ....	79
Figura 38: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Identificación de los Datos.....	80
Figura 39: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Información del Contenido.....	80
Figura 41: Diagrama Colaboración. CU: Consultar Metadatos por Palabras Claves.....	81

## Tablas

Tabla 1: Descripción del actor del negocio .....	39
Tabla 2: Descripción del trabajador del negocio.....	39
Tabla 3: Descripción textual del CUN: Consultar Metadatos Geológicos.....	41
Tabla 4: Descripción del actor del sistema. ....	46
Tabla 5: Descripción textual del CU: Consultar Metadatos por Palabras Claves .....	47
Tabla 6: Descripción textual del CU: Consultar Metadatos por Categorías .....	50
Tabla 7: Descripción textual del CU: Mostrar Núcleo del Metadato.....	51
Tabla 8: Descripción textual del CU: Mostrar el Metadato Completo .....	51
Tabla 9: Descripción textual del CU: Salvar el Metadato. ....	52

### **Introducción**

La informática es la disciplina que estudia el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores[1], es decir, es la encargada de realizar sistemas de softwares capaces de gestionar información.

Desde su nacimiento, hace casi un siglo, ha contribuido al desarrollo acelerado de la humanidad, debido al sinnúmero de aplicaciones que tiene en todas las ramas del conocimiento, las que se han potenciado aún más con el surgimiento de Internet y la posibilidad que esta brinda de compartir la información.

Son innumerables los beneficios que brinda la informática, lo que implica que el mundo entero este inmerso en la asimilación y utilización de dicha tecnología. Es por ello que Cuba apuesta hoy al desarrollo de la misma, con el énfasis puesto en la informatización de la sociedad e identificando la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones; y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible.[2]

Entre los proyectos que se han llevado a cabo para la materialización de la informatización de la sociedad en nuestro país, está la creación de la Universidad de las Ciencias Informáticas (UCI), universidad que vincula dinámicamente el estudio con la producción y la investigación, acorde con las necesidades sociales del país y de otros pueblos hermanos[3].

La UCI lleva a cabo una serie de proyectos que contribuyen al desarrollo de las distintas esferas de la sociedad. Uno de ellos lo constituye el Programa Nacional para la Informatización del Conocimiento Geológico, que se realiza en conjunto con la Oficina Nacional de Recursos Minerales, con el objetivo de informatizar las principales funciones de dicha entidad.

La Oficina Nacional de Recursos Minerales (ONRM) es la autoridad minero petrolera de la República de Cuba, encargada de administrar el conocimiento geológico y toda la información geológica, minera y petrolera de la nación. Dicha información se encuentra almacenada en distintas bases de datos diseminadas por todo el territorio nacional y en diferentes formatos, provocando esto que se dificulte y sea ineficiente la recuperación de la misma referente a la calidad y ubicación de los datos.

Para documentar toda la información existente y la que se cree en un futuro se hace uso de los metadatos, estructura que proporciona una mejor organización y localización de los datos. Por lo que

se hace necesario contar con un sistema que sea capaz de realizar la consulta y recuperación de estos.

Por este motivo se propone realizar la investigación a partir del siguiente problema científico: *Inexistencia en la Oficina Nacional de Recursos Minerales de un software que permita la recuperación de los metadatos geológicos.*

Teniendo en cuenta el problema anterior se declara como objeto de esta investigación: *El proceso de recuperación de los metadatos geológicos.*

Por tanto, se constituye en objetivo de este trabajo *Diseñar una aplicación Web que permita la recuperación de los metadatos geológicos.*

Asumiendo este objetivo general, el campo de acción de esta investigación resulta: *La informatización de los procesos de recuperación de los metadatos geológicos en la Oficina Nacional de Recursos Minerales.*

Teniendo en cuenta lo anterior se plantea como idea a defender que: *Con el análisis del proceso de recuperación de información en la Oficina Nacional de Recursos Minerales se podrá diseñar un sistema capaz de automatizar el proceso de recuperación de los metadatos.*

Para el cumplimiento de la investigación serán desarrolladas las siguientes tareas científicas:

- Estudio del estado del arte del tema tratado en la investigación.
- Estudio de cómo ocurre el proceso de recuperación de información en la Oficina Nacional de Recursos Minerales.
- Estudio de sistemas para la creación de metadatos de datos geológicos.
- Estudio de las tecnologías para el desarrollo del software.
- Modelación de los procesos a informatizar.
- Diseño de la Base de datos donde se almacenará la información referente a los metadatos.
- Análisis y Diseño de la aplicación Web para la consulta de metadatos geológicos en la Oficina Nacional de Recursos Minerales.

Para el desarrollo de estas tareas se emplearán como métodos científicos:

### Métodos Teóricos

- Análisis y Síntesis para el estudio de la bibliografía y demás información obtenida relacionada con el objeto de estudio.
- Modelación para modelar los procesos del negocio en la entidad y realizar una reproducción simplificada de la realidad.

### Métodos Empíricos

- Entrevista individual realizada al Ing. César Rosales y al Ing. Héctor Rodríguez con el fin de recopilar toda la información necesaria sobre el proceso de recuperación de los metadatos geológicos en la Oficina Nacional de Recursos Minerales y definir las capacidades o funcionalidades del sistema a desarrollar.
- Observación para realizar un registro visual de lo que ocurre en el entorno del problema, aportar nuevos elementos que puedan ser de interés científico.

La estructura del Trabajo de Diploma contiene:

Capítulo 1. Fundamentación Teórica: Se describe la situación actual de la Oficina Nacional de Recursos Minerales y de los procesos automatizar, se exponen los principales problemas que motivan el desarrollo de esta investigación, se describe brevemente en qué consiste la propuesta y se hace referencia a otros sistemas automatizados existentes.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar: Se realiza un estudio referente a las tendencias y tecnologías existentes en la actualidad que han sido seleccionadas para la confección del proyecto, planteándose lo fundamentado en cada caso.

Capítulo 3. Presentación de la solución propuesta: Se describen los procesos actuales a través de un modelo de negocio, el cual sirve de base para determinar qué se va a desarrollar. Se determinan las funcionalidades del sistema propuesto y se describen en detalle.

Capítulo 4. Construcción de la solución propuesta: Aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de análisis y de diseño y se plantea el modelo de datos.

# Capítulo 1

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En este capítulo serán expuestos conceptos como metadatos, metadatos geográficos y recuperación de información, los que conformarán el objeto de estudio que se llevará a cabo durante la investigación.

Se analiza el uso de metadatos como solución a la recuperación de información en la Oficina Nacional de Recursos Minerales (ONRM), la importancia de estos como una estructura bien organizada para la localización de dicha información y por último se exponen las características principales de algunos sistemas a nivel internacional que proveen servicios de metadatos.

El contenido de este capítulo tiene como objetivo principal sentar las bases teóricas de la investigación para una mejor comprensión del lector.

### 1.2 Conceptos asociados al dominio del problema

La Geología es la ciencia que se encarga del estudio del subsuelo (composición, estructura, evolución, recursos minerales), utiliza las diferentes teorías que explican la evolución física del planeta (causas) para:

- Dar uso correcto a los recursos minerales y las aguas subterráneas.
- Manejar la peligrosidad sísmica y el riesgo geológico.
- Evaluar los suelos para la construcción.

Esta ciencia basa sus métodos de estudio en una serie de aproximaciones sucesivas expresada por la escala del estudio partiendo de lo más general a lo particular, desde el planeta y sus grandes estructuras (continentes) hasta tipo de roca o mineral como objeto de la investigación.

Tales investigaciones consisten en la observación de las estructuras en superficie o descubiertas mediante excavaciones o perforaciones (laboreos), la realización de ensayos (químicos, mineralógicos, petrográficas, físico mecánicos), el estudio de las variaciones (globales o locales) de los campos físicos naturales o inducidos de la Tierra (gravitatorio, magnético, sísmicos, eléctricos) que se miden mediante equipos tecnológicos para ese fin. Toda esa información tiene en común la ubicación espacial (georeferenciada) expresada en coordenadas. De esta manera la base y resultados de las



investigaciones geológicas están expresados por datos de textos (las observaciones), mediciones discretas, registros continuos o imágenes.

Tanto la información de partida como los resultados de esas investigaciones sobre la estructura y la composición del subsuelo de un territorio dado (Cuba y su zona económica, en este caso) constituye el conocimiento geológico.

El conjunto de instituciones autorizadas para la generación, administración y uso fundamental de este conocimiento se les denomina entidades geológicas.

En resumen, los trabajos del servicio geológico producen datos que es la forma en que se materializa el conocimiento geológico. Esos datos pueden ser primarios (observaciones, mediciones) o conclusiones producto del procesamiento de los datos primarios (informes, mapas).[4] Una vez digitalizados u organizados como producto de la ejecución de un proyecto de conformación de base de datos, se les considera datos geográficos digitales.[5]

Los datos geográficos digitales pretenden modelar y describir el mundo real para su posterior análisis y visualización mediante medios muy diversos. Sus características principales así como sus limitaciones deben estar completamente documentadas mediante los metadatos.[6]

### **1.2.1 Metadatos. Metadatos Geográficos**

Definición: El término metadatos está compuesto por el prefijo inseparable de origen griego *meta* que, según el Diccionario de la Lengua Española significa "junto a", "después de", "entre" o "con" y el término de origen latino *datum* (lo que se da), que según esta misma fuente tiene dos acepciones significativas: la primera: "antecedente necesario para llegar al conocimiento exacto de algo o para deducir las consecuencias legítimas de un hecho"; y la segunda: "información dispuesta de manera adecuada para su tratamiento por un ordenador".[1]

Si se analiza esta definición, puede entenderse que los metadatos constituyen datos que describen un conjunto de información o como comúnmente se les denomina: datos sobre datos.

Para tener una mejor comprensión de qué se entiende por metadato, a continuación se hace referencia a conceptos de algunos autores respecto al tema:

Para Heery y Dempsey (1998): "Los metadatos son datos que se asocian con objetos que liberan a sus usuarios potenciales (personas o programas) de la necesidad de tener un conocimiento avanzado y completo sobre su existencia o características (...). Los metadatos constituyen un conocimiento que permite a los usuarios, humanos o automatizados, comportarse de manera inteligente".[7]

Méndez (2002), por su parte, considera que definir los metadatos requiere tener en cuenta las diferentes tendencias de éstos, con el fin de limitar el concepto y acercarlo más hacia lo que realmente representan: “El término metadatos, alude a cualquier dato que ayuda en la identificación, descripción de recursos electrónicos en red. Son pues, datos que describen los atributos de un recurso, caracterizan sus relaciones, soportan su recuperación, gestión y uso efectivo, y existen en un contexto estrictamente electrónico”. [8]

Daudinot (2006) cuando se refiere a los metadatos expresa que: “... son datos secundarios como pueden ser el autor, el título, las palabras clave, el resumen, la fecha, u otros que describen los datos primarios o recursos de información, es decir, se emplean para suministrar información sobre datos producidos, ellos describen el contenido y otras características de los datos primarios para posibilitar a una persona o máquina ubicar y entender los datos.” [9]

Por su parte Oliva y Quezada (2006) plantean que: “Los datos que conforman un metadato generalmente dan respuesta a las preguntas quién, qué, cuándo, cómo, dónde y porqué. En el metadato usualmente se recoge información sobre cada una de las etapas de la existencia de los datos que se documentan, así como de su semántica, aspecto vital para lograr un uso adecuado del mismo.” [10]

De estas definiciones puede extraerse una conclusión y es que, el término metadatos, se refiere a cualquier dato que ayude a las personas a identificar, gestionar y valorar la calidad de un conjunto de información, sin tener que consultarla completamente.

En la actualidad el concepto de metadatos se está convirtiendo en una herramienta de gran ayuda para aquellas personas que trabajan con información espacial, debido principalmente al gran volumen que existe de dicha información y la necesidad de su organización y recuperación.

¿Qué se entiende entonces por metadatos geográficos?

Tal como plantea Nebert (2004), los metadatos geográficos ayudan a las personas involucradas en el uso de información geográfica a encontrar los datos que necesitan y a determinar la mejor manera de cómo usarlos. [6]

Por su parte Oliva(2006) considera que: Un metadato geográfico es un conjunto de información que identifica diferentes aspectos relacionados a grupos de datos o a datos específicos y permite conocer características de esos datos que los particularizan dentro de un conjunto. Describe aspectos de los datos geoespaciales como son: calidad, actualización, referencia geoespacial, autor, entre otras.[10]

Se puede concluir entonces que, los metadatos geográficos constituyen información sobre la forma y contenido de los datos espaciales.

El Comité Federal de Datos Geográficos (2000) reconoce que la creación de metadatos geográficos persigue tres objetivos (y a su vez beneficios) principales:[6]

- Organizar y mantener la inversión en datos hecha por una organización: los metadatos buscan fomentar la reusabilidad de datos sin tener que recurrir al equipo humano que se encargó de su creación inicial. Aunque la creación de metadatos parezca un coste adicional, el valor de los datos a la larga es dependiente de su documentación.
- Publicitar la existencia de información geográfica a través de sistemas de catálogo: mediante la publicación de recursos de información geográfica a través de un catálogo, las organizaciones pueden encontrar datos a usar, otras organizaciones con las que compartir datos y esfuerzos de mantenimiento y clientes para esos datos. En general, permiten a los usuarios utilizar los datos de un modo más eficiente, determinando si serán de utilidad para ellos.
- Proporcionar información que ayude a la transferencia de los datos: los metadatos deberían acompañar siempre a los propios datos. Facilitan el acceso a los datos, su adquisición y una mejor utilización de los datos logrando una interoperabilidad de la información cuando esta procede de fuentes diversas. Los metadatos ayudan al usuario u organización que los recibe en el procesamiento, interpretación, y almacenamiento de los datos en repositorios internos.

### **1.2.2 Recuperación de información**

En la actualidad la recuperación de información (RI) ha adquirido un rol muy importante, debido principalmente, al valor que tiene la misma. Se puede plantear que disponer o no de la información justa en tiempo y forma puede resultar en el éxito o fracaso de una operación.

¿Qué se entiende concretamente por recuperación de información?

Croft (1987) estima que la recuperación de información es: “el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado. En estas tareas desempeñan un papel fundamental los lenguajes documentales, las técnicas de resumen, la descripción del objeto documental, entre otras”. [11]

Por otro lado, Korfhage (1997) definió la recuperación de información como “la localización y presentación a un usuario de información relevante a una necesidad de información expresada como una pregunta”. [12]

Según Robertson y Spark-Jones (1977) : “La recuperación de información engloba las acciones encaminadas a identificar, seleccionar y acceder a los recursos de información útiles al usuario, sin perjuicio de otras acepciones del concepto, en las que puede profundizarse utilizando la bibliografía correspondiente”. [13]

Se puede considerar a partir de aquí que la recuperación de información no es más que el proceso que da respuesta a la solicitud de un usuario sobre determinado recurso.

### **1.3 La Oficina Nacional de Recursos Minerales y el proceso de recuperación de información**

La Oficina Nacional de Recursos Minerales (ONRM) es la autoridad minera de la República de Cuba, como institución con personalidad jurídica, adscrita al Ministerio de la Industria Básica, es la entidad encargada de: [14]

- Fiscalizar y controlar la actividad minera y el uso racional de los recursos minerales, asesorando al Ministerio de la Industria Básica en esta materia, y a los demás organismos de la Administración Central del Estado, sin perjuicio de sus respectivas competencias.
- Aprobar, registrar y controlar las reservas minerales, certificando el grado de preparación de los yacimientos para su asimilación industrial.
- Emitir los dictámenes técnicos sobre el otorgamiento, anulabilidad y extinción de concesiones mineras, y fiscalizar el cumplimiento de las condiciones bajo las cuales se otorgó la concesión.
- Aprobar los proyectos de explotación minera.
- Llevar el Registro Minero y mantener actualizadas las anotaciones sobre concesiones mineras, áreas mineras reservadas, yacimientos, manifestaciones minerales, áreas en investigación y minas en explotación o abandonadas.
- Constituirse en depositario de la información geológica y minera de la nación.
- Ejercer la inspección estatal sobre las personas naturales y jurídicas que ejecuten la actividad minera, para comprobar el cumplimiento de los acuerdos y obligaciones a que se hayan comprometido dichas entidades, así como de las disposiciones legales vigentes que rijan la actividad que se inspecciona.
- Controlar la ejecución de los planes de preservación del medio ambiente y de las medidas para mitigar el impacto ambiental.
- Mantener actualizadas las estadísticas mineras del país.

- Participar en el cierre de minas y controlar las medidas del programa de cierre que se ejecuten.

La Oficina Nacional de Recursos Minerales tiene una estructura de un solo nivel, su oficina central, que tiene las direcciones que a continuación se relacionan:[4]

- Dirección General.
- Dirección de Documentación e Informática.
- Dirección de Control Económico.
- Dirección de Registro Minero.
- Dirección de Evaluación Económica.
- Dirección de Minerales.
- Dirección de Níquel.
- Dirección de Hidrocarburos.

De manera general la ONRM es la entidad geológica encargada de administrar y gestionar el conocimiento geológico obtenido a lo largo de más de 40 años. Uno de los principales problemas que afectan en la actualidad el funcionamiento de dicha entidad, es la recuperación de información geológica, pues esta se encuentra distribuida por todo el territorio nacional y en distintos formatos, tornándose difícil la localización de la misma a la hora de satisfacer determinado interés, lo que entorpece su desarrollo, comercialización e intercambio.

Generalmente localizar un dato geoespacial que se necesite para un fin determinado se torna muy difícil, pues no existe un mecanismo lo suficientemente general que difunda las características fundamentales de los datos existentes, así como su disponibilidad.[10]

Dentro de un volumen inmenso de información, como el que dispone hoy en día la Oficina Nacional de Recursos Minerales, es muy engorroso hacer una búsqueda que satisfaga un interés concreto. Por ejemplo: determinar qué dato o conjunto de datos cumplen con cierta condición es muy complicado, porque sería necesario consultarlos todos y para eso tendría que ser un especialista en la materia que además tuviera conocimiento del formato en que estos están almacenados.

Otro problema es que no siempre es posible tener alguna medida de la calidad de la información que se tiene. Esto tiene sus causas en el hecho de que la información referente a la calidad no forma parte del dato, por lo que es una cuestión imprecisa valorarla a través de un análisis del mismo.[15]

Se necesita entonces una estructura bien organizada que permita documentar fácilmente la información existente y la que se cree en un futuro; de forma tal que se puedan hacer búsquedas en dicha estructura, publicar características fundamentales de esta y hacer sobre ella análisis sin gran dificultad.[10] Por lo que se propone catalogar los datos haciendo uso de los metadatos.

Los metadatos en el mundo, constituyen un paso de gran importancia para el almacenamiento y gestión del conocimiento. Entre sus principales funciones están las siguientes: [16]

- Organizar los datos para conseguir que los usuarios los conozcan mejor y, en consecuencia, los utilicen, evitando duplicaciones y detectando errores. Todo esto ayuda a mantener la inversión en los datos.
- Facilitar la transferencia de los datos al aportar información sobre formatos de archivos, volumen, localización, etc.
- Permitir una distribución en línea de los datos incorporando direcciones para descargar archivos, ya sean gratuitamente o previo pago, mediante unas descripciones de cómo realizar estos procesos.
- Facilitar la búsqueda de datos en una compleja base de datos corporativa. Los metadatos permiten a un usuario hacer una consulta, que es redireccionada a los diferentes catálogos registrados en el servidor principal. El resultado es un listado de metadatos procedentes de diferentes servidores.
- Evitar el uso erróneo de los datos al incluir descripciones de cómo se crearon, con qué finalidad y cómo se deben utilizar.
- Asegura la integridad de los datos así como la seguridad de los mismos.

De esta manera cuando un usuario necesita consultar información acerca de determinado dato geológico dirige su solicitud a la ONRM, esta verifica la identidad del usuario, si el mismo esta autorizado para acceder a la información requerida, la ONRM consulta el catálogo y otorga la información.

### **1.3.1 Importancia del uso de Metadatos de Información Geográfica. [10]**

Los metadatos tienen como funcionalidad práctica fundamental ayudar a los usuarios a discernir si la información es la más adecuada para un fin concreto sin tener que consultarla directamente. El usuario

podrá obtener elementos para la identificación de la información buscada a través de palabras claves que caracterizan la información geográfica.

Los metadatos son imprescindibles en la construcción de una sólida infraestructura de datos geoespaciales y su utilización facilita la documentación de los datos.

El uso de metadatos trae ventajas como son:

- Una mejor organización de la institución o compañía y con ello mejores resultados.
- Mantiene el valor de la inversión en datos geográficos. Hace persistente el conocimiento acerca de los datos cuando el personal calificado que los creó o que tiene relación directa con ellos y conoce todas sus características, deja de prestar servicios en la institución.
- Brinda información sobre los datos disponibles de manera que posibilita su catalogación, así como facilita información de los lugares y formas de almacenamiento de dichos datos.
- Provee información rápida a potenciales comercializadores de los datos geográficos.
- Mediante la información que ofrecen es posible el procesamiento de archivos provenientes de fuentes no conocidas o ajenas al usuario que está consultando.
- Facilita la ubicación y la detección de los datos, para así lograr identificarlos, localizarlos, accederlos y que puedan ser utilizados por personas o instituciones que los requieran.
- Provee una guía de los datos geográficos en cuanto a sus rasgos principales.
- Después que los datos de interés han sido localizados guía la interpretación y el uso de los mismos.
- Posibilita un mejor y mayor intercambio de datos entre organizaciones. De esta manera es mucho más claro qué se ofrece y qué se requiere, lo que facilita la cooperación y el trabajo coordinado inter-institucional acerca de temas geoespaciales.
- Propicia un mecanismo para mantener y auditar los datos existentes.
- Facilita la comunicación y el intercambio entre el personal especializado y el mundo exterior. Fortalece las relaciones empresa-sociedad.
- Previene la duplicación de la información, porque facilita verificar si existe un dato con características similares.

### 1.3.2 Estándares para la documentación del metadato[16]

Para que los metadatos se puedan compartir entre los diferentes productores y usuarios, deben ser comprensibles por todos ellos. Para ello, se han diseñado los estándares de metadatos que pretenden poner en común los puntos de vista de los distintos actores implicados en la información espacial para que esta documentación sea comprensible para todos, es decir, para que todos “hablemos el mismo idioma”.

La *International Organization for Standardization* (ISO) es una federación de organismos nacionales de estandarización. Esta organización está dividida, internamente en comités técnicos encargados de elaborar estándares para diferentes temáticas. El comité destinado a la información geográfica es el ISO/TC 211, que elaboró el estándar para metadatos geológicos ISO 19115. Además de este estándar, existe una extensa lista de otros estándares relacionados con la información geográfica.

El estándar ISO 19115 tiene como principal objetivo proporcionar una estructura definitiva y única para describir datos geográficos digitales. Este estándar define los elementos de los metadatos, proporciona un esquema y establece un conjunto común de terminología de metadatos y definiciones. La implementación de este estándar:

- Proporcionará a los productores de los datos la información necesaria para describir correctamente sus datos.
- Facilitará la organización y el mantenimiento de los metadatos de datos geográficos.
- Permitirá a los usuarios utilizar los datos geográficos de una forma más eficiente, porque conocerán sus características básicas.
- Facilitará el descubrimiento, la recuperación y la reutilización de los datos. A los usuarios les será más fácil localizar, acceder, evaluar, adquirir y utilizar los datos.

Una parte de dicho estándar atiende específicamente el caso de las imágenes *ráster* (ISO19115-2: “*Geographic Information- Metadata for imagery and gridded data*”), debido a las necesidades que su proceso de producción tiene de ser documentado para mantener los controles de calidad en los productos finales.

Por otro lado hay que mencionar también la importancia del estándar ISO19139: “*Geographic Information –Metadata- XML schema implementation*”, que define cómo se deben codificar los metadatos ISO19115 sobre XML (formato de exportación de metadatos).



Finalmente, el *ISO Core* comprende un conjunto de metadatos del estándar ISO19115 que son considerados como más fundamentales o importantes en la documentación de los datos espaciales. Se trata con ello de permitir la posibilidad de acortar la tarea de catalogación de metadatos, que sería muy extensa si se atiende a los 401 ítems del estándar, muchos de los cuales son demasiado específicos o difíciles de conocer.

Los estándares ofrecen una base a través de la cuál pudieran desarrollarse perfiles nacionales o más específicos para una materia o interés determinado.[10] Para la realización de este proyecto se utilizará el Procedimiento para la Documentación del metadato, desarrollado por la empresa Geominera Oriente, cuyo análisis de contenido y modelado conceptual fueron realizados precisamente siguiendo la norma ISO 19115. Este procedimiento en sí no constituye un perfil de la norma debido a que el Comité técnico cubano constituido para estos fines trabaja aún en su análisis para su posterior aprobación.

#### **1.4 Análisis de otras soluciones existentes**

A continuación se exponen algunos sistemas de información geológica que utilizan metadatos para la gestión de datos geológicos en distintos países. En Cuba no existen soluciones implementadas de este tipo.

##### Sistema de Información Geológico Minero de Extremadura (SIGEO)[17]

El SIGEO es un sistema de difusión vía Internet de la información geológica y minera georeferenciada de Extremadura (España) que tiene como objetivo el proveer libre acceso a toda la información sobre los recursos geológicos y mineros, todo ello con el fin último de promover y fomentar la inversión minera en la región.

El Sistema de Información Geológico-Minero de Extremadura provee acceso interactivo a los datos geoespaciales correspondientes a la geología; hidrogeología; recursos hidrominerales; indicios mineros; sondeos y calicatas de investigación; teledetección; campañas geoquímicas; geofísica; estudios, campañas y proyectos de rocas y minerales industriales; muestras de rocas y minerales industriales, afloramientos de interés de rocas ornamentales.

El sistema está subdividido en tres subconjuntos: en primer lugar, la cartografía interactiva; en segundo lugar, los informes de los proyectos de investigación; y, en tercer lugar, los metadatos de los mapas interactivos.

Para acceder a los metadatos se ha desarrollado un buscador, en el que se han diferenciado 13 categorías. Eligiendo una categoría determinada aparecen todas las capas de información geográfica de esa categoría y los iconos correspondientes para acceder a sus metadatos. También se puede acceder a los metadatos desde los mapas temáticos.

### Geodata [18]

Geodata se crea en 1999 por Javier Alcalá y Guillem Terradas para atender la demanda de servicios de GIS y de expertos en tecnologías de la información geográfica, principalmente por parte de la administración pública catalana.

Geodata ofrece servicios y elabora productos relacionados con la información territorial como son:

- Integración de toda la información cartográfica y documental asociada.
- Facilitar la comprensión del territorio.
- Mejorar la gestión y planificación territorial.
- Divulgar información de interés público.
- Maximizar la velocidad y potencia de consulta.
- Mejorar la atención y servicio al ciudadano.
- Automatizar tareas administrativas.
- Optimizar recursos y hacer más ágil el trabajo al personal técnico.
- Dar soporte a la toma de decisiones, tanto a nivel técnico como político.
- Minimizar costes en la adquisición de licencias de software GIS..
- Disponer de soporte informático adicional y de nuevas tecnologías.
- Añade además metadatos para las cartografías.

### Instituto Geológico y Minero de España (IGME) [19]

IGME ofrece servicios como:

- Centro de Documentación.
- Biblioteca.
- Servicio de Publicaciones.

- Precios públicos.
- Centro Nacional de Datos Polares.
- Centro Nacional de Referencias de Suelos.
- Oficina de transferencia de resultados de investigación.

Además tiene habilitado un Portal en el *Antarctic Master Directory* para poder realizar consultas sobre los metadatos españoles, utilizando los parámetros propios del formato de Intercambio de Datos (DIF).

#### Instituto Geológico Minero y Metalúrgico (INGEMMET) [20]

INGEMMET, ha elaborado sus metadatos basados en el estándar Internacional ISO 19115-19139 *Geographic Information Metadata*, constituyéndose en el primer nodo de una red distribuida, conectada electrónicamente, de productores, administradores y usuarios de información geográfica (*Clearinghouse*).

Servicios que ofrece.

- Catálogos de Metadatos ISO19115:2002 & 19139-2007.
- Servicio Web Map Server (WMS). ISO 19128.
- Servicio de *ImageServerArcMap* en IMS.
- Servicio de Cobertura en KML.

#### Australian Spatial Data Directory (ASDD) [21]

El Directorio de Datos Espaciales Australiano (ASDD) proporciona interfaces de búsqueda para descubrir descripciones de conjuntos de datos geospaciales (metadato) en todas partes de Australia. Una descripción de un conjunto de datos es un documento conciso que coherentemente explica un cierto juego de datos geospaciales y provee información adicional sobre los datos reales.

### **1.5 Conclusiones**

Con el análisis de los conceptos expuestos anteriormente y el estudio de los diferentes sistemas de recuperación de información geológica que proveen servicios de metadatos en distintas partes del mundo, se concluye que la solución es construir una aplicación que se ajuste a las condiciones de nuestro país y de la ONRM, debido a que la norma ISO 19115 solo provee una base para la documentación del metadato geológico y a partir de ahí cada país debe ajustarla a sus intereses y características geográficas.

## Capítulo 2

### Tendencias y tecnologías actuales a desarrollar

#### 2.1 Introducción

En este capítulo se muestra una visión general de las tendencias y tecnologías actuales; los lenguajes de programación, los sistemas gestores de bases de datos, la metodología, así como otras herramientas que se utilizarán para la construcción del sistema propuesto.

#### 2.2 Aplicaciones Web

La rápida expansión de Internet, ha supuesto la necesidad de que toda la información que se maneja en esta sea accesible desde cualquier sitio y en el momento que sea requerida, es por ello que en la actualidad se está produciendo un enorme auge en la creación de aplicaciones Web.

Farre (2005) plantea que: “En ingeniería de software, una aplicación Web es una aplicación ubicada en un servidor Web que ofrece un determinado servicio y que es accesible desde diversas redes telemáticas, siendo la más común Internet (World Wide Web). Los usuarios acceden a la aplicación utilizando un navegador Web y no es necesario descargarse ningún tipo de software adicional ya que la aplicación se ejecuta remotamente”. [22]

Concluyendo una aplicación Web es aquella a la que el cliente accede a través de Internet comunicándose con un servidor en el que se ejecuta la aplicación.

##### 2.2.1 Características de las aplicaciones Web

En una aplicación web hay varias partes implicadas: la parte servidora, la parte cliente y la infraestructura para establecer la comunicación entre estas. Los usuarios utilizan la parte cliente, sobre la que pueden interactuar y que permite generar peticiones a través de Internet, mediante el protocolo web HTTP, a la parte servidora, que tras realizar su cometido devolverá la respuesta al cliente.

Las aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor.[23]

La arquitectura de una aplicación Web además incluye la aplicación en el servidor, que es la que permite al sistema manejar lógica de negocio. [24]

Las aplicaciones Web suelen estar estructuradas siguiendo un modelo basado en tres niveles:

Nivel de presentación: es el encargado de generar la interfaz de usuario en función de las acciones llevadas a cabo por el mismo.

Nivel de negocio: contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario.

Nivel de administración de datos: encargado de hacer persistente toda la información, suministra y almacena información para el nivel de negocio.

Los dos primeros y una parte del tercero (el código encargado de las actualizaciones y consultas), suelen estar en el servidor mientras que la parte restante del tercer nivel se sitúa en la base de datos (notar que, debido al uso de procedimientos almacenados en la base de datos, una parte del segundo nivel también puede encontrarse en la misma).[25]

En general las características básicas de una aplicación Web son las siguientes:

- Está alojada en un servidor Web y sigue una arquitectura cliente/servidor.
- Es accesible a través de una red telemática y mediante la utilización de un navegador Web.
- La lógica de la aplicación Web se ejecuta íntegramente en el servidor Web y el navegador del cliente sólo representa los datos.
- La navegación a través de ella y la entrada de datos por parte de un usuario afectan el estado de la lógica de la aplicación.
- Puede interactuar con otros procesos y aplicaciones Web.
- El acceso a la aplicación puede ser público o restringido, dependiendo hacia quien vaya dirigida.
- Las actualizaciones y el mantenimiento de la aplicación es transparente para los usuarios, ya que estos no deberán instalar ni actualizar nada.
- Las aplicaciones son multiplataforma ya que pueden ejecutarse en cualquier Sistema Operativo que posea un navegador Web.

### **2.3 Arquitectura de software**

En el campo del software, la arquitectura ha evolucionado como una disciplina de gran importancia, su adecuada representación y diseño permite entender el sistema, organizar su desarrollo, plantear la reutilización del software y hacerlo evolucionar, convirtiéndose de esa manera en lo que se conoce como el esqueleto de soporte del sistema.

¿Qué se entiende entonces por Arquitectura de Software (AS)?

Una definición reconocida es la de Clements[26]: La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema.

Por su parte Booch, Jacobson y Rumbaugh en su libro El Proceso Unificado de Desarrollo de Software, plantean que:

“Arquitectura es la estructura de los componentes más significativos de un sistema interactuando a través de interfaces con otros componentes conformados por componentes sucesivamente pequeños e interfaces”. [27]

Otras de las definiciones más utilizadas es la que brinda el documento de IEEE Std 1471-2000, adoptada también por Microsoft:

"La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución." [28]

Es mucha la abundancia de definiciones en el campo de la Arquitectura de Software y aunque ninguna de ellas es respaldada unánimemente por la totalidad de los arquitectos, existe un acuerdo general en que esta se refiere a la estructura a grandes rasgos del sistema, estructura consistente en componentes y relaciones entre ellos. [28]

### **2.3.1 Arquitectura de una Aplicación Web: Modelo Cliente/Servidor**

En una aplicación Web la arquitectura es sencilla, estas siguen el modelo cliente/servidor, modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

IBM define al modelo Cliente/Servidor como:

"La tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores" [29].

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes.

¿Qué es el cliente?

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente[29].

¿Qué es el servidor?

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.[29].

Una representación gráfica de este tipo de arquitectura sería la siguiente:

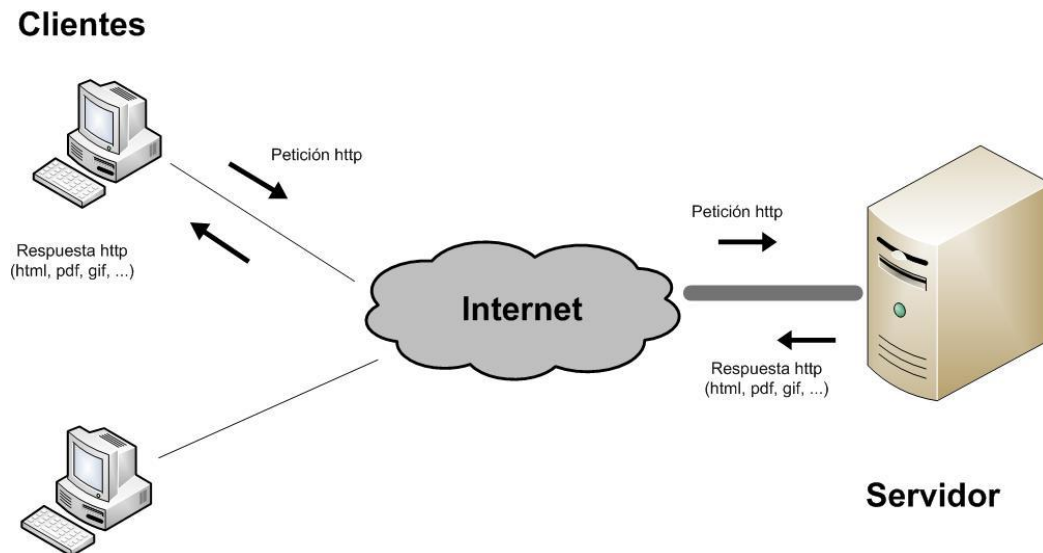


Figura 1: Arquitectura clásica cliente/servidor

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes[30]:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.

- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### 2.3.2 Modelo – Vista – Controlador (MVC)

Reconocido como estilo arquitectónico por Taylor y Medvidovic, considerado una micro-arquitectura por Robert Allen y David Garlan. En ocasiones se lo define más bien como un patrón de diseño o como práctica recurrente, y en estos términos es referido en el marco de la estrategia arquitectónica de Microsoft. En la documentación correspondiente es tratado a veces en términos de un estilo decididamente abstracto y otras como patrón de aplicación ligado a una implementación específica. Buschmann y otros lo consideran un patrón correspondiente al estilo de los sistemas interactivos[31].

MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones[32]:

- Modelo: Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- Vista: Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- Controlador: Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio (*service requests*) para el modelo o la vista.

#### Ciclo de vida MVC

El ciclo de vida del MVC es representado por las 3 partes mencionadas anteriormente y el cliente (también conocido como usuario). El siguiente diagrama representa el ciclo de vida de manera sencilla:



Figura 2: Ciclo de vida MVC



El ciclo de vida empieza cuando el usuario hace una solicitud al Controlador con información sobre lo que desea realizar. El Controlador decide a quien debe delegar la tarea y es aquí donde el Modelo empieza su trabajo. En esta etapa el Modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el Controlador. Una vez que termina su labor, le regresa al Controlador la información resultante de sus operaciones, el cual a su vez dirige a la Vista. La Vista se encarga de transformar los datos en información visualmente entendible al usuario.

Entre las principales ventajas de hacer uso del MVC están[31]:

- Soporte de vistas múltiples. Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos, mostrado de maneras diferentes.
- Adaptación al cambio. Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.
- Permite la escalabilidad de la aplicación en caso de que sea requerido. El diseño MVC tiene una estructura organizada y debido a la clara separación de tareas que posee, facilita la modificación para aumentar las capacidades funcionales o de almacenamiento de la aplicación, asegurando la independencia del desarrollo hecho a los cambios sufridos.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

## 2.4 Los diferentes lenguajes de programación para la Web

Los lenguajes de programación para la web se dividen en dos grupos: los del lado del servidor y los del lado del cliente.

González (2007) plantea que:[33]

- Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Esta no se verá bien si el ordenador cliente no tiene instalados los *plug-in* adecuados. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y ello puede afectar a la seguridad.

- Un lenguaje del lado servidor es independiente del navegador utilizado, que no necesitará *plug-in* especiales para visualizar correctamente cualquier página. Además, el código de los script puede ocultarse al terminal cliente, que sólo verá el código HTML terminado.

Concluyendo los lenguajes del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y se envían al cliente a través de Internet en un formato comprensible para él. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas. Los lenguajes del lado del cliente son los que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

#### **2.4.1 Lenguaje de programación del lado del servidor: PHP**

PHP es el acrónimo de *Hipertext Preprocesor*. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Entre sus principales características se pueden mencionar:

- Multiplataforma: PHP funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra.
- Licencia de software libre: PHP es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, ni está limitada su distribución y, es posible ampliarlo con nuevas funcionalidades si así se desea.
- Sintaxis cómoda: PHP cuenta con una sintaxis similar a la de C, C++ o Perl.
- Extensa librería de funciones: PHP cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores.
- Compatibilidad con bases de datos: Amplio soporte para una gran cantidad de bases de datos. Tiene acceso a un gran número de gestores de bases de datos: Adabas D, dBase, Empress, Ingress, InterBase, FrontBase, DB2, Informix, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase, etc.

## 2.4.2 Lenguaje de programación del lado del cliente

### HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

El código HTML, indica básicamente donde insertar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. Este no es más que una serie de etiquetas que se utilizan para definir la forma o estilo que se quiere aplicar al documento.

Un documento HTML ha de estar delimitado por la etiqueta <html> y </html>. Dentro de este documento, se pueden distinguir dos partes principales:

El encabezado, delimitado por <head> y </head> donde se colocan las etiquetas de índole informativo como por ejemplo el título de la página.

El cuerpo, flanqueado por las etiquetas <body> y </body>, que será donde se coloque el texto e imágenes delimitados a su vez por otras etiquetas.

Entre sus principales ventajas se pueden destacar las siguientes:

- Sencillo, permite describir hipertexto.
- Presenta texto de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

## 2.4.3 Fundamentación del lenguaje de programación a utilizar

Para la implementación de esta aplicación, se utilizará como lenguaje de programación PHP en su versión 5.0. Para justificar esta elección se han tenido en cuenta las ventajas que ofrece.

Ventajas PHP 5.0:

- Su gratuidad, independencia de plataforma, rapidez y seguridad.
- Se caracteriza por ser un lenguaje muy rápido y fácil de aprender.
- Soporta la orientación a objeto.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

### **2.5 AJAX: Nueva técnica para el desarrollo de aplicaciones Web**

El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML".

Pérez (2008) define AJAX de la siguiente forma[34]:

“AJAX no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

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.

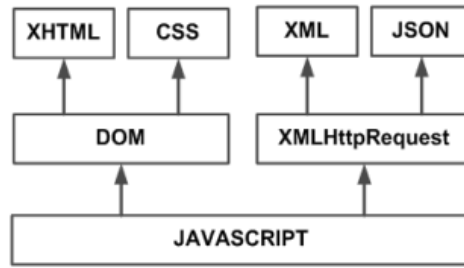


Figura 3: Tecnologías agrupadas bajo el concepto de AJAX.

Aunque la técnica tradicional para crear aplicaciones Web funciona correctamente, no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, pues el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

En el siguiente esquema se muestra una comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX.

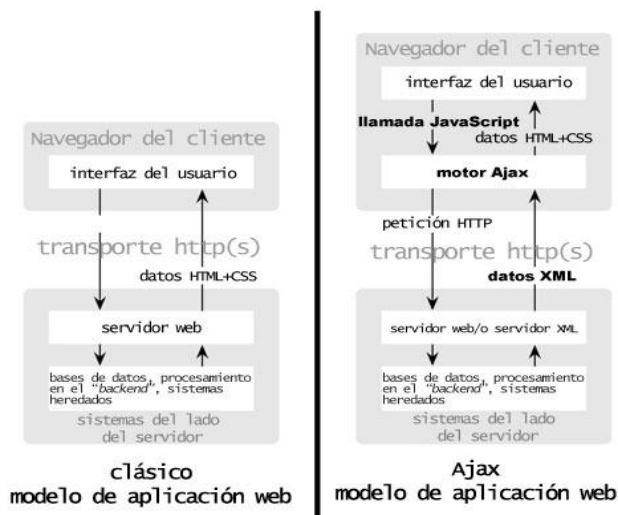


Figura 4: Comparación entre AJAX y el modelo tradicional de aplicación web.

Razones para usar AJAX[35]:

- Basado en los estándares Abiertos: AJAX está formado por las tecnologías JavaScript, HTML, XML, CSS, y XML HTTP Request Object, y soportado por los navegadores más utilizados de internet como son: Mozilla, Internet Explorer, Safari y Opera.
- Usabilidad: Permite a las páginas hacer una pequeña petición de datos al servidor y recibirla sin necesidad de cargar la página entera.
- Es independiente del tipo de tecnología de servidor que se utilice: Así como AJAX funciona en cualquier navegador, es perfectamente compatible con cualquier tipo de servidor estándar y lenguaje de programación Web.
- Beneficia las aplicaciones web: AJAX es la cara del presente en las aplicaciones web, las aplicaciones web conllevan ciertos beneficios sobre las aplicaciones de escritorio. Esto incluye un menor coste de creación, facilidad de soporte y mantenimiento, menores tiempos a la hora de desarrollarlas, y sin necesidad de instalaciones. AJAX solo ayudará a las aplicaciones web a mejorar y conseguir un mejor resultado de cara al usuario final.
- No es difícil su utilización: Como AJAX está basada en los estándares utilizados comúnmente, significa que no es un gran esfuerzo el aprendizaje de las tecnologías que esta requiere, el pasar de un simple código HTML y aplicaciones Web a una potente aplicación AJAX, no es de gran dificultad. También significa que los desarrolladores pueden actualizar poco a poco las interfaces de usuario hacia unas interfaces con AJAX y no necesitan una re-escritura de la aplicación entera, se puede hacer incrementalmente.

**2.6 ¿Qué es un *framework*?**

Según plantea Gutiérrez, en el desarrollo de software, un *framework* es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Agrega funcionalidad extendida a un lenguaje de programación; ésta, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito, proporciona una estructura al código y hace que los desarrolladores escriban código mejor, más entendible y mantenible. Además hace la programación más fácil, convirtiendo complejas funciones en sencillas instrucciones. Está usualmente escrito en el lenguaje que extiende.[36]

Un *framework* permite separar en capas la aplicación. En general, divide la aplicación en tres capas:

- La lógica de presentación que administra las interacciones entre el usuario y el software.
- La lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.
- La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.

### 2.6.1 *Framework* para PHP: **Symfony**

Symfony está desarrollado completamente con PHP5. 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.

Características:

Symfony se diseñó para que se ajustara a los siguientes requisitos [44]:

- 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 solo 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 adaptables 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.

Symfony automatiza la mayoría de elementos comunes de los proyectos Web, como por ejemplo:

- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.

- La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del *framework*. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
- Los formularios incluyen validación automatizada y relleno automático de datos (“repopulation”), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- El soporte de correo incluido y la gestión de *APIs* permiten a las aplicaciones Web interactuar más allá de los navegadores.
- Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- Los *plug-ins*, las factorías (patrón de diseño “Factory”) y los *mixins* permiten realizar extensiones a medida de Symfony.
- Las interacciones con AJAX son muy fáciles de implementar mediante los *helpers* que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código.

Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software:

- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- El *framework* de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas (*test-driven development*).
- La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.



- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- Es posible realizar cambios “en caliente” de la configuración (sin necesidad de reiniciar el servidor).
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

### 2.6.2 ¿Por qué Symfony?

La utilización de este *framework* provee grandes ventajas como son:

- Symfony sólo funciona con PHP 5, lo que garantiza el máximo rendimiento y permite aprovechar PHP hasta el límite.
- Es escalable hasta cualquier límite (siempre que se dispongan de los recursos técnicos necesarios).
- Es software libre, con licencia MIT, con la que se pueden desarrollar aplicaciones libres o comerciales.
- Dispone de una de las mejores documentaciones del mundo del software libre, ya que aúna cantidad, calidad, gratuidad y una actualización continua.
- La programación en Symfony es sencilla, pues simplifica al máximo la aplicación de patrones de diseño y buenas prácticas propias de los mejores programadores.
- Incorpora los últimos avances en interacción con usuarios (como por ejemplo AJAX).

Concluyendo, Symfony, es un framework maduro, bien documentado y con una gran comunidad que lo apoya.

## 2.7 Servidores web

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP).

### 2.7.1 Apache

El servidor Web Apache es un servidor Web gratuito desarrollado por el *Apache Server Project* (Proyecto Servidor Apache) cuyo objetivo es la creación de un servidor Web fiable, eficiente y

fácilmente extensible con código fuente abierto gratuito.

Ventajas:

- Su licencia. Esta es de código abierto y permite el uso comercial y no comercial de Apache.
- Una talentosa comunidad de desarrolladores siguiendo un proceso abierto de desarrollo.
- Arquitectura modular. Los usuarios de Apache pueden adicionar fácilmente funcionalidad a sus ambientes específicos.
- Portabilidad. Apache trabaja sobre todas las versiones recientes de Unix y Linux, Windows, BeOs.
- Es robusto y seguro.

## 2.8 ¿Qué es un Sistema Gestor de Bases de Datos?

Mato(1999) plantea que, un Sistema Gestor de Base de Datos (SGBD ) es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.[37].

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

Los programas de aplicación operan sobre los datos almacenados en la base de datos utilizando las facilidades que brindan los SGBD, los que, en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios.

### 2.8.1 PostgreSQL como Sistema Gestor de Base de Datos

PostGreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley, es una derivación libre (*OpenSource*) de este proyecto, y utiliza el lenguaje SQL92/SQL99. Al ser sistema objeto-relacional, incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostGreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Las principales características de este gestor de bases de datos son:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora la estructura de datos arreglos.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

### **2.8.2 Fundamentación del Sistema Gestor de Base de Datos a utilizar**

Entre las principales ventajas que proporciona este SGBD están:

- Instalación ilimitada: PostgreSQL no tiene costo asociado a la licencia del software, por lo que puede ser instalado en cualquier parte.
- Ahorros considerables en costos de operación: PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
- Extensible: El código fuente está disponible para todos sin costo. Si se necesita extender o personalizar PostgreSQL, puede hacerse con un mínimo esfuerzo, sin costos adicionales.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada Acceso Concurrente Multiversión, para conseguir una mejor respuesta en ambientes de grandes volúmenes.

- Herramientas gráficas de diseño y administración de BD: Tiene varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).
- Posee una gran escalabilidad: Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- Implementa el uso de rollback's, subconsultas y transacciones: Lo que hace su funcionamiento mucho más eficaz.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos: Equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

## 2.9 El Proceso Unificado de Desarrollo de Software como metodología de desarrollo.

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. De forma general una metodología en un proyecto de desarrollo de software define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

### El Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software (RUP, llamado así por sus siglas en inglés *Rational Unified Process*) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes.

En su modelación se definen como principales elementos:

Trabajadores (“quién”): define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de actividades (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

RUP divide en 4 fases el desarrollo del software y agrupa las actividades en grupos lógicos definiendo 9 flujos de trabajo principales. Los 6 primeros conocidos como flujos de ingeniería y los tres últimos como de apoyo.

### Fases

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias versiones del producto que han pasado las pruebas. Se ponen estas versiones a consideración de un subconjunto de usuarios.
- **Transición:** La versión ya está lista para su instalación en las condiciones reales. Puede implicar reparación de errores.

### Flujos de trabajo

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y Diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.

- **Instalación:** Produce versiones del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

En la figura que se muestra a continuación se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

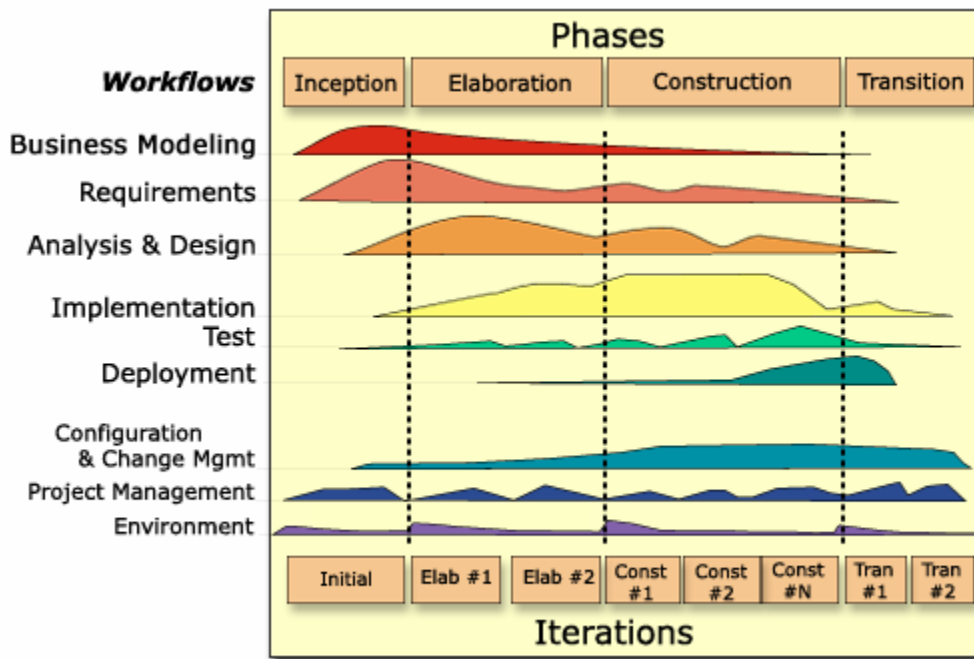


Figura 5: RUP en dos dimensiones.

El ciclo de vida de RUP se caracteriza por ser:

- Dirigido por casos de uso: Los casos de uso (CU) reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

### 2.9.1 ¿Por qué RUP?

Existen varias razones que respaldan esta elección, la primera de ellas es que RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración y describe los requerimientos de la aplicación desde el punto de vista del usuario.

RUP es un proceso pesado, basado mucho en la documentación. Existen diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.) con los que se controla el desarrollo del software. A través de un predefinido esquema de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos o corregirlos. RUP define en cada momento del ciclo de vida del proyecto, que artefactos, con qué nivel de detalle, y por qué rol se deben crear. Se definirán qué artefactos son necesarios para realizar una actividad y qué artefactos se deberán crear durante dicha actividad.

Con RUP se presentarán al cliente los artefactos del final de una fase y se valorarán las precondiciones para la siguiente (definición de riesgos, aceptación del plan de iteración, prototipos) y solo después de que el cliente acepte los artefactos generados se pasará a la siguiente fase. La

calidad de los artefactos generados será probada durante la totalidad del ciclo de vida del proyecto a través de distintas medidas de calidad.

RUP genera también versiones basadas en los artefactos después de cada fase, pero en su caso no se limitan solo al código, sino que estas vienen acompañadas de todo lo que traería el producto final.

### **2.10 El Lenguaje Unificado de Modelado como soporte de la modelación de la solución propuesta.**

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés *Unified Model Language*) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades del modelo, en un entorno de desarrollo complejo. Contiene construcciones para representar decisiones de implementación y para elementos de tiempo de ejecución en componentes.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

### **2.11 Otras herramientas a utilizar**

#### **2.11.1 Visual Paradigm**

Visual Paradigm 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. Permite realizar ingeniería tanto directa como inversa. Además, la herramienta es colaborativa, es



decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, y permite control de versiones.

Visual Paradigm ofrece:

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

### 2.11.2 Eclipse

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

El IDE de Eclipse es un entorno de desarrollo libre para crear aplicaciones clientes de cualquier tipo. El proyecto Eclipse es un proyecto de desarrollo de software de código abierto dedicado a proporcionar una plataforma industrial robusta, con amplias características y con calidad comercial para el desarrollo de herramientas altamente integradas.

Eclipse está formado por su propio núcleo y por multitud de *plug-ins* que yacen sobre éste. Este tipo de arquitectura basada en *plug-ins* es la que le proporciona una flexibilidad y escalabilidad muy elevadas. Todo componente con funcionalidad es tratado como un *plug-in* en Eclipse. El SDK de Eclipse incluye la plataforma básica y dos herramientas principales que resultan de utilidad para el desarrollo de *plug-ins*.

Pese a que Eclipse esté escrito en su mayor parte en Java (salvo el núcleo), se ejecuta sobre la máquina virtual de ésta y su uso más popular es como un IDE para Java. Eclipse es neutral y adaptable a cualquier tipo de lenguaje, por ejemplo C/C++, Cobol, C#, XML, etc.

## **2.12 Conclusiones**

Después de haber realizado un estudio de las tecnologías a utilizar en el sistema propuesto, así como de algunos conceptos y tendencias actuales, se arribó a la conclusión de que para el desarrollo de la aplicación se utilizará como lenguaje de programación PHP en su versión 5.0 y como sistema gestor de bases de datos PostGreSQL 8.2. La aplicación será hosteada en un servidor Web Apache y se utilizará como lenguaje del lado del cliente HTML. Como metodología de desarrollo de software se escogió RUP y la modelación de todo el sistema de llevará a cabo con la ayuda de la herramienta case Visual Paradigm. Se utilizarán otras herramientas que complementarán la propuesta de solución, Eclipse en su versión 3.2 como IDE y como framework para PHP: Symfony versión 1.0.

## Capítulo 3

### PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

#### 3.1 Introducción

En el capítulo se presenta la modelación del negocio propuesto, se definen los actores, trabajadores y casos de uso correspondientes, así como la descripción textual de cada uno de ellos. Se especifican los requerimientos funcionales y no funcionales que permitirán el desarrollo de la aplicación y se describe la solución propuesta en términos de casos de uso del sistema.

#### 3.2 Modelo del Negocio propuesto

Se realiza un modelamiento del negocio con el objetivo de comprender la estructura y la dinámica de la Oficina Nacional de Recursos Minerales, organización en la cual se va a implantar el sistema, así como para comprender los problemas actuales de la misma e identificar las mejoras potenciales.

##### 3.2.1 Actores y Trabajadores del Negocio

El término actor del negocio se refiere a cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.[38]

Actor	Descripción
Cliente	Persona o institución interesada en obtener determinada información geológica

*Tabla 1: Descripción del actor del negocio*

Un trabajador del negocio representa a personas o sistemas (software) dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso. Son los que en un futuro se convertirán en usuarios del sistema que se quiere construir. [38]

Trabajador	Descripción
Especialista	Persona encargada de suministrar la información solicitada por el usuario.

*Tabla 2: Descripción del trabajador del negocio*

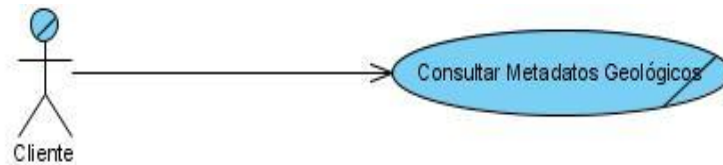
### 3.2.2 Procesos del Negocio

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos.[38]

El proceso de negocio que es objeto de estudio en la investigación es la consulta de metadatos geológicos en la Oficina Nacional de Recursos Minerales.

### 3.2.3 Diagrama de Casos de Uso del Negocio

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.[38]



*Figura 6: Diagrama de Caso de Uso del Negocio.*

### 3.2.4 Descripción textual de los Casos de Uso del Negocio

Con el objetivo de tener una mejor comprensión acerca del proceso que se lleva a cabo en el negocio, a continuación se muestra la descripción textual del caso de uso del negocio y con el mismo propósito se presenta el diagrama de actividades, artefacto que describe gráficamente el flujo de las actividades que tiene lugar en la realización del caso de uso.

**Descripción textual del caso de uso**

<b>Caso de uso del negocio</b>	<b>Consultar Metadatos Geológicos</b>
Actor	Cliente (inicia)
Trabajador	Especialista
<p><b>Resumen</b></p> <p>El caso de uso inicia cuando un cliente llega a la Oficina Nacional de Recursos Minerales y solicita consultar determinada información, el especialista de la oficina realiza las verificaciones pertinentes y en dependencia de estas otorga o no la información solicitada.</p>	
Casos de uso asociados	-
<b>Flujo de trabajo</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
<p>1. El caso de uso inicia cuando el cliente llega a la ONRM y solicita consultar determinada información.</p> <p>2. Entrega los documentos de identificación.</p> <p>3. Recibe la información y se retira.</p>	<p>1.1 El especialista solicita los documentos de identificación.</p> <p>2.1 Verifica los documentos de identificación.</p> <p>2.2 Verifica el tipo de información solicitada.</p> <p>2.3 Verifica si el cliente está autorizado a recibir la información solicitada.</p> <p>2.4 El especialista procede a localizar la información.</p> <p>2.5. Entrega la información solicitada.</p>
<b>Cursos alternos</b>	
3.1 El cliente se retira.	2.4.1 Si no esta autorizado el cliente para consultar la información solicitada, se le notifica.

*Tabla 3: Descripción textual del CUN: Consultar Metadatos Geológicos.*

**Diagrama de actividades**

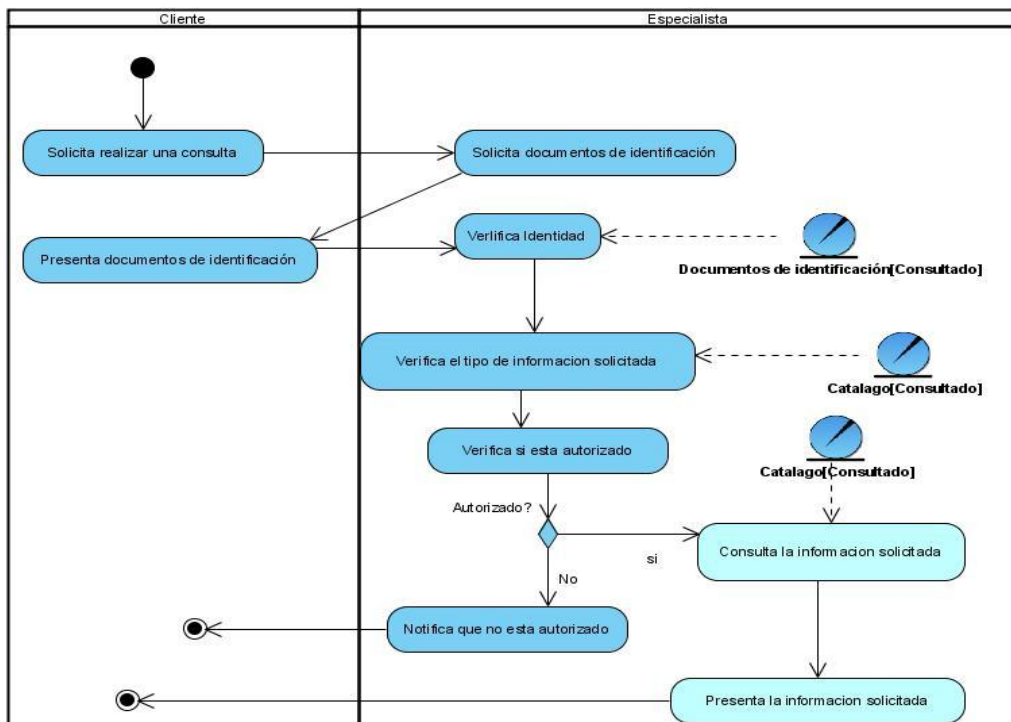


Figura 7: Diagrama de actividades del negocio.

**3.2.5 Modelo de Objetos del Negocio**

El Modelo de Objetos del Negocio muestra la participación de los trabajadores (Especialista) y entidades<sup>1</sup> del negocio (Catálogo y Documentos de Identificación) y la relación entre ellos.

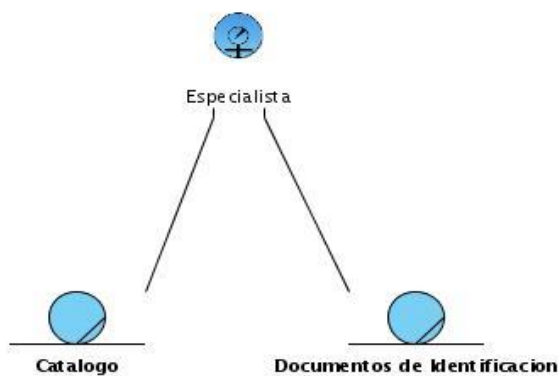


Figura 8: Modelo de Objeto.

<sup>1</sup> Representan un contenedor de información, algo físico que se utiliza en el proceso del negocio y que sirve para obtener información o actualizar información.

### **3.3 Requerimientos Funcionales/No Funcionales del sistema**

En este epígrafe se listan los requerimientos funcionales y no funcionales que debe cumplir el software.

Requerimientos Funcionales (RF): Son capacidades o condiciones que el sistema debe cumplir.[39]

Requerimientos no Funcionales (RNF): Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son características que hacen al producto atractivo, usable, rápido o confiable. [39]

#### **3.3.1 Requerimientos Funcionales**

RF1. El sistema debe permitir consultas por Palabras Claves.

RF2. El sistema debe permitir consultas por Categorías.

2.1 El sistema debe permitir consultas en la sección Información del Metadatos.

2.2 El sistema debe permitir consultas en la sección Identificación de los Datos.

2.3 El sistema debe permitir consultas en la sección Información del Contenido.

2.4 El sistema debe permitir consultas en la sección Calidad de los Datos.

RF3. El sistema deberá ofrecer la posibilidad de mostrar el núcleo de un metadato seleccionado.

RF4. El sistema deberá ofrecer la posibilidad de mostrar la totalidad de un metadato seleccionado.

RF5. El sistema deberá ofrecer la posibilidad de salvar en un archivo XML el metadato seleccionado.

#### **3.3.2 Requerimientos No Funcionales**

##### Requerimientos de Apariencia o Interfaz externa

La interfaz del sistema será diseñada de forma sencilla y fácil de usar, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Será formal, seria y con una navegación sugerente, teniendo en cuenta el fin con el que se desarrolla la aplicación. Se integrará con la técnica de desarrollo AJAX para lograr una aplicación interactiva. Además deberá conjugar con la apariencia del portal.

##### Requerimientos de Rendimiento

La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios del sistema sea en un período de tiempo breve (de segundos) para evitar la

acumulación de trabajo por parte de los responsables. El sistema deberá de ser lo más estable y confiable posible. El servicio siempre debe mantenerse activo y con opciones de recuperación ante cualquier fallo de conexión.

### Requerimientos de Portabilidad

La aplicación Web podrá ser usada bajo cualquier versión del sistema operativo Linux, para su implementación se emplearon herramientas de programación y gestión de bases de datos que pertenecen a Software Libre. Además debe permitir que pueda ser consultado desde cualquier plataforma.

### Requerimientos de Soporte

Garantía de instalación y prueba del sistema. Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra. Además debe brindar manuales de instalación y de soporte técnico. Si se obtienen nuevas versiones o actualizaciones se implantarán.

### Requerimientos de Software

La PC que funcionará como servidor de aplicaciones deberá contar con:

- Sistema operativo: Linux
- Servidor Web: Apache.

La PC que funcionará como servidor de base de datos deberá contar con:

- Sistema operativo: Linux
- Gestor de Base de Datos: PostGreSQL 8.2.

Las PCs que funcionarán como cliente podrán contar con cualquier sistema operativo y navegador web.

### Requerimientos de Hardware

El servidor de aplicaciones deberá contar con:

- Procesador Pentium IV o superior.
- 1 Gb de memoria RAM o superior.
- 40 Gb o superior de Disco Duro.

El servidor de base de datos deberá contar con:



- Procesador Pentium IV o superior.
- 1 Gb de memoria RAM o superior.
- 120 Gb o superior de Disco Duro.

Las PCs clientes podrán contar como mínimo con un cliente ligero.

#### Restricciones en el diseño e implementación

- Se utilizará PHP 5.x como lenguaje de programación
- Se utilizará Symfony como framework de desarrollo.
- Se utilizará XML como lenguaje de transporte.
- Se utilizará como entorno de desarrollo Eclipse.
- Se utilizará como herramienta CASE el Visual Paradigm.

#### Requerimientos Legales

El sistema se basa en un estándar que se rige por normas internacionales y cumple con las normas y leyes establecidas en nuestro país y a nivel internacional. La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL. Las normas ISO utilizadas son: ISO 19115 y 19139 regidas por los estándares internacionales para la documentación del Metadato.

#### Requerimientos de disponibilidad

El sistema deberá estar disponible las 24 horas todos los días del año y soportar un gran número de conexiones simultáneas.

#### Requerimientos de Procedimientos

El sistema debe contar con manuales de procedimientos que aseguren el buen funcionamiento de los RF del sistema.

### **3.4 Descripción del Sistema propuesto**

La propuesta de solución es la creación de una aplicación Web que permita la consulta de metadatos de las bases de datos que existen en la Oficina Nacional de Recursos Minerales, el Instituto de Geología y Paleontología, CUPET y GeoMinsal, facilitando de esta manera para los usuarios la recuperación de los datos geológicos.

Esta aplicación será hosteada en un servidor Web Apache, se utilizará como lenguaje de programación PHP, y el gestor de base de datos para almacenar la información será PostGreSQL.

### 3.4.1 Descripción de los actores

Los actores del sistema son los trabajadores del negocio que tiene actividades automatizar, si algún actor del negocio interactúa con el sistema entonces también será un actor del sistema, es decir, representan terceros fuera del sistema que colaboran con este.

Actor	Descripción
Usuario	Persona o institución interesada en realizar una consulta.

Tabla 4: Descripción del actor del sistema.

### 3.4.2 Casos de Usos del Sistema

Con el objetivo de lograr un mejor entendimiento de la solución propuesta, a continuación se muestra el diagrama de casos de uso del sistema<sup>2</sup>, así como la descripción textual de cada caso de uso.

#### 3.4.2.1 Diagrama de casos de uso del sistema

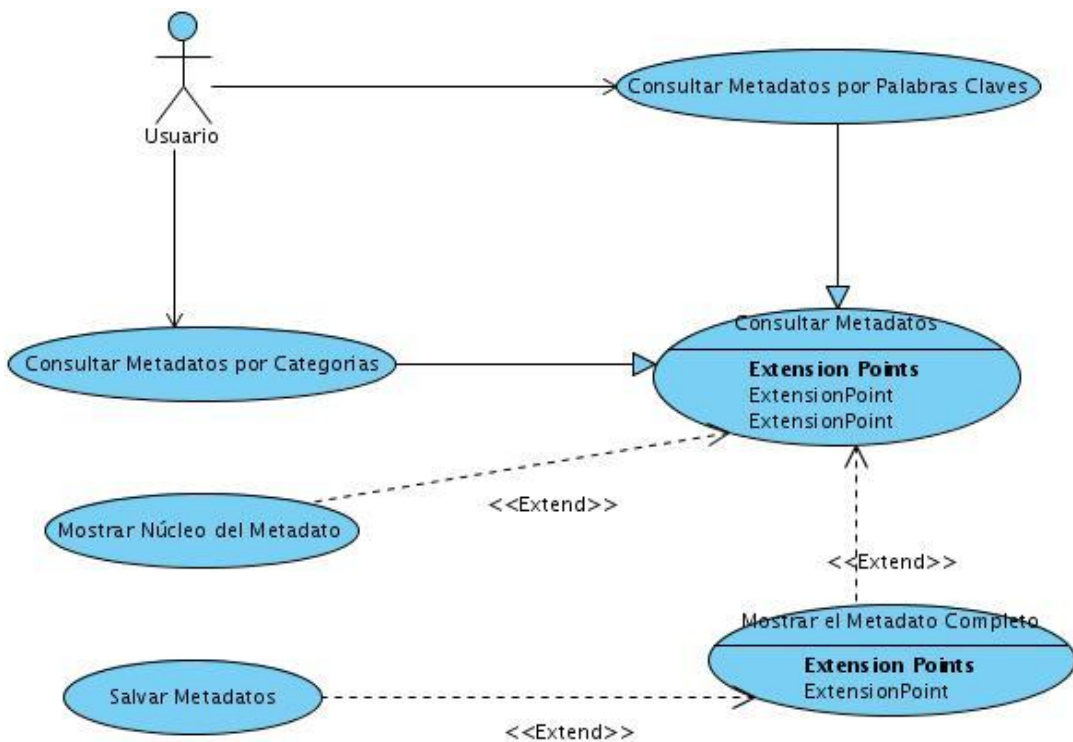


Figura 9: Diagrama de Casos de Uso del Sistema

<sup>2</sup> Representa gráficamente a los procesos y su interacción con los actores.

3.4.2.2 Descripción de los casos de uso del sistema

**CU \_ Consultar Metadatos por Palabras Claves**

<b>Caso de Uso:</b>	Consultar Metadatos por Palabras Claves.	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario escoge la opción Metadatos, el sistema muestra la interfaz consultar metadatos por palabras claves, el usuario introduce las palabras por las que desea realizar la búsqueda y se muestra la consulta realizada.	
<b>Precondiciones:</b>		
<b>Referencias</b>	RF1	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1- El usuario escoge la opción Metadatos, en el Portal de la ONRM.	1.1 El sistema muestra la interfaz para la consulta de metadatos por palabras claves.
	2- El usuario introduce las palabras por las que desea realizar la búsqueda.	2.1- El sistema busca los metadatos que contengan al menos una de las palabras claves. 2.2 - El sistema muestra una lista con el resultado de la consulta.
<b>Poscondiciones</b>	-	

*Tabla 5: Descripción textual del CU: Consultar Metadatos por Palabras Claves*

**CU \_ Consultar Metadatos por Categorías.**

<b>Caso de Uso:</b>	<b>Consultar Metadatos por Categorías</b>
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario escoge la opción de Búsqueda Avanzada, el sistema muestra las opciones Información del Metadato, Identificación de los Datos,

	Información del Contenido y Calidad de los Datos y en dependencia de la que se seleccione se mostrará un formulario donde el usuario debe introducir los datos necesarios para que el sistema muestre la consulta.
<b>Precondiciones:</b>	-
<b>Referencias</b>	RF2, RF2.1, RF2.2, RF2.3, RF2.4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario escoge la opción de Búsqueda Avanzada	1.1- El sistema muestra un formulario con las opciones Información del Metadato, Identificación de los Datos, Información del Contenido y Calidad de los Datos.
2- El usuario escoge uno de las siguientes opciones: a) Información del Metadato. b) Identificación de los Datos. c) Información del Contenido. d) Calidad de los Datos.	2.1- En caso de haber escogido la opción: a) Ver Sección Información del Metadato. b) Ver Sección Identificación de los Datos. c) Ver Sección Información del Contenido. d) Ver Sección Calidad de los Datos
<b>Sección Información del Metadato.</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario accede al formulario para la consulta en la sección Información del Metadato.	1.1- El sistema muestra la interfaz para la consulta en la sección Información del Metadato y solicita los datos necesarios para la búsqueda.

<b>Sección Identificación de los Datos</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario accede al formulario para la consulta en la sección Identificación de los Datos.	1.1- El sistema muestra la interfaz para la consulta en la sección Identificación de los Datos y solicita los datos necesarios para la búsqueda.
2- El usuario introduce al menos uno de los datos solicitados.	2.1- El sistema busca los metadatos que contengan los datos especificados 2.2- El sistema muestra una lista con el resultado de la consulta.
<b>Sección Información del Contenido</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario accede al formulario para la consulta en la sección Información del Contenido.	1.1- El sistema muestra la interfaz para la consulta en la sección Información del Contenido y solicita los datos necesarios para la búsqueda.
2- El usuario introduce al menos uno de los datos solicitados.	2.1- El sistema busca los metadatos que contengan los datos especificados 2.2- El sistema muestra una lista con el resultado de la consulta.

<b>Sección Calidad de los Datos</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario accede al formulario para la consulta en la sección Calidad de los Datos.	1.1- El sistema muestra la interfaz para la consulta en la sección Calidad de los Datos y solicita los datos necesarios para la búsqueda.
2- El usuario introduce al menos uno de los datos solicitados.	2.1- El sistema busca los metadatos que contengan los datos especificados 2.2- El sistema muestra una lista con el resultado de la consulta.
<b>Poscondiciones</b>	-

Tabla 6: Descripción textual del CU: Consultar Metadatos por Categorías

**CU \_ Mostrar el Núcleo del Metadato.**

<b>Caso de Uso:</b>	<b>Mostrar el Núcleo del Metadato</b>
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el usuario escoge la opción Ver Núcleo de uno de los metadatos mostrados en la lista del caso de uso Consultar Metadatos, el sistema muestra el núcleo del metadato seleccionado.
<b>Precondiciones:</b>	- El usuario debe tener seleccionado un metadato de la lista en el caso de uso Consultar Metadatos.
<b>Referencias</b>	RF3
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario escoge la opción Ver Núcleo.	1.1- El sistema busca el núcleo del metadato escogido.

	1.2- El sistema muestra el núcleo del metadato seleccionado.
<b>Poscondiciones</b>	-

Tabla 7: Descripción textual del CU: Mostrar Núcleo del Metadato

**CU \_ Mostrar el Metadato Completo.**

<b>Caso de Uso:</b>	<b>Mostrar el Metadato Completo</b>	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso inicia cuando el usuario escoge la opción Ver Completo de uno de los metadatos mostrados en la lista del caso de uso Consultar Metadatos, el sistema muestra el metadato seleccionado.	
<b>Precondiciones:</b>	- El usuario debe tener seleccionado un metadato de la lista en el caso de uso Consultar Metadatos.	
<b>Referencias</b>	RF4	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	1- El usuario escoge la opción Ver completo.	1.1- El sistema busca el metadato escogido.
		1.2- El sistema muestra completamente el metadato seleccionado.
<b>Poscondiciones</b>	-	

Tabla 8: Descripción textual del CU: Mostrar el Metadato Completo

**CU \_ Salvar el Metadato**

<b>Caso de Uso:</b>	<b>Salvar el Metadato.</b>	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario escoge la opción salvar en el caso de uso Mostrar Metadato Completo, el sistema permite salvar el fichero del metadato.	
<b>Precondiciones:</b>	- Debe haberse mostrado el metadato que se desea salvar.	
<b>Referencias</b>	RF5	
<b>Prioridad</b>	secundario	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	1- El usuario escoge la opción salvar.	1.1- El sistema muestra un formulario para especificar el lugar donde será salvado el fichero así como el nombre del mismo
	2- El usuario especifica el lugar y el nombre del fichero.	2.1- El sistema exporta en formato XML el contenido del metadato seleccionado y lo salva.
<b>Poscondiciones</b>		

*Tabla 9: Descripción textual del CU: Salvar el Metadato.*

**3.5 Conclusiones**

En el capítulo se realizó un estudio detallado de los procesos que se llevan a cabo en la Oficina Nacional de Recursos Minerales y que constituyen el objeto de estudio para la realización de esta aplicación, identificando los problemas que presenta el negocio y determinando cuáles casos de usos pasarán a ser automatizados.

El desarrollo de este flujo de trabajo dará paso al diseño del sistema, comenzando así la construcción de la solución propuesta.



## Capítulo 4

### CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

#### 4.1 Introducción

En este capítulo se abordan los resultados principales del flujo de trabajo Análisis y Diseño e Implementación. La construcción de la solución propuesta se realiza a partir del refinamiento de los requisitos tanto funcionales como no funcionales. Se especifican los principales artefactos que se generan durante estos flujos de trabajo (Modelo de Análisis y Diseño), así como el diseño de la base de datos.

Con el objetivo de obtener una mejor descripción de la aplicación que se desarrolla se definen además el Modelo de Implementación y Despliegue.

#### 4.2 Análisis

Durante este flujo de trabajo, se analizan los requisitos que se describieron en la primera etapa del desarrollo del software, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requerimientos y una descripción de los mismos, que sea fácil de mantener y que ayude a estructurar el sistema entero (incluyendo la arquitectura).[40]

Concluyendo el análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales[40], esbozando cómo llevar a cabo la funcionalidad dentro del sistema. Sirve como una primera aproximación al diseño.

El lenguaje que se utiliza en el Análisis se basa en un modelo de objetos conceptual, que se denomina modelo de análisis (Ver Anexo I. Diagramas de Clases del Análisis).

##### 4.2.2 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Consisten en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.[41]

Existen dos tipos de diagramas de interacción en UML.

- Diagrama de colaboración: Se muestra la interacción entre los objetos, creando enlaces entre ellos y añadiendo mensajes a esos enlaces. Enfatizan la organización estructural de los objetos que participan en una interacción (Ver Anexo II. Diagramas de Colaboración.)

- Diagrama de secuencia: Muestra las secuencias de tiempo y destaca la ordenación temporal de los mensajes.

### 4.3 Diseño

El modelo de diseño es un 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 programación, tienen impacto en el sistema a considerar, siendo una entrada fundamental para las actividades de implementación.[42]

#### 4.3.1 Diagramas de clases del diseño

En la presente investigación se hace una propuesta para el diagrama de clases del diseño basándose en la implementación que realiza Symfony<sup>3</sup> de la arquitectura MVC, quedando estructurada de la siguiente manera:

El Controlador: Se divide en un controlador frontal y en las acciones.

Controlador frontal: Ofrece un punto de entrada único para toda la aplicación, controla todas las peticiones de la misma, carga la configuración y determina la acción a ejecutarse.

Acciones: Incluyen el código específico del controlador de cada página y pueden incluir la lógica de la aplicación cuando esta no es muy compleja. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación. Utilizan el modelo y definen variables para la vista.

Las acciones son métodos con el nombre *executeNombreAccion* de una clase llamada *nombreModuloActions* que hereda de la clase *sfActions* y se encuentran agrupadas por módulos.

Para el diseño de la aplicación se utiliza una sintaxis alternativa, que distribuye las acciones en archivos separados, un archivo por acción, proporcionando mayores facilidades para el trabajo en equipo y la utilización de herramientas de control de versiones. Para la implementación en este caso, cada clase acción extiende *sfAction* (en lugar de *sfActions*) y su nombre es *nombreAccionAction*. El nombre del método es simplemente *execute*.

La Vista: Se separa en un layout y en una plantilla.

Layout: Es global en toda la aplicación o al menos en un grupo de páginas y contiene el código HTML común para estas.

---

<sup>3</sup> Framework que se utiliza para el desarrollo de la aplicación.

Plantilla: Son la presentación de los datos de la acción que se está ejecutando y se encarga de visualizar las variables definidas en el controlador.

El modelo: Se divide en la capa de acceso a datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas (se explicará con más detalle en el siguiente epígrafe).

En la figura que se muestra a continuación se describe el flujo de trabajo de Symfony, basándose en lo anteriormente explicado:

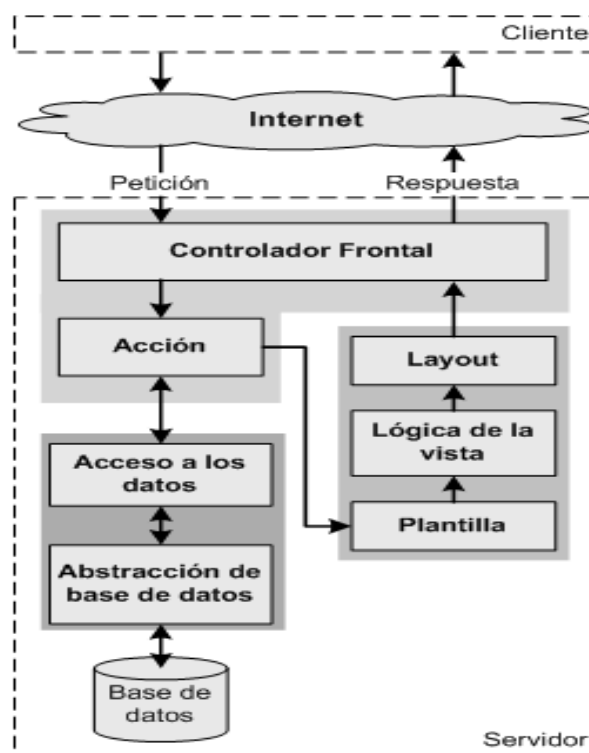


Figura 10: Flujo de trabajo de Symfony.

Cuando un cliente realiza una solicitud, esta es recibida por el controlador frontal (representado por index.php en los diagramas de clases del diseño), que se encarga de descodificarla y mediante los mecanismos de enrutamiento del framework seleccionar la acción que se ejecutará para dar respuesta a la petición. La acción hace uso del modelo, si es necesario, para iniciar las variables que se mostrarán.

Los componentes internos del framework identifican la plantilla que corresponde a la acción que se ejecuta, esta le da formato a las variables modificadas en la acción y luego es decorada con el layout. Finalmente el controlador frontal construye la respuesta a la petición y la envía al cliente.

Una vez analizado cómo ocurre el flujo de trabajo en Symfony y aplicando lo anteriormente expuesto, se presenta el diseño de la aplicación en términos de diagramas de clases dividido por casos de uso, con el objetivo de proporcionar una mejor comprensión.

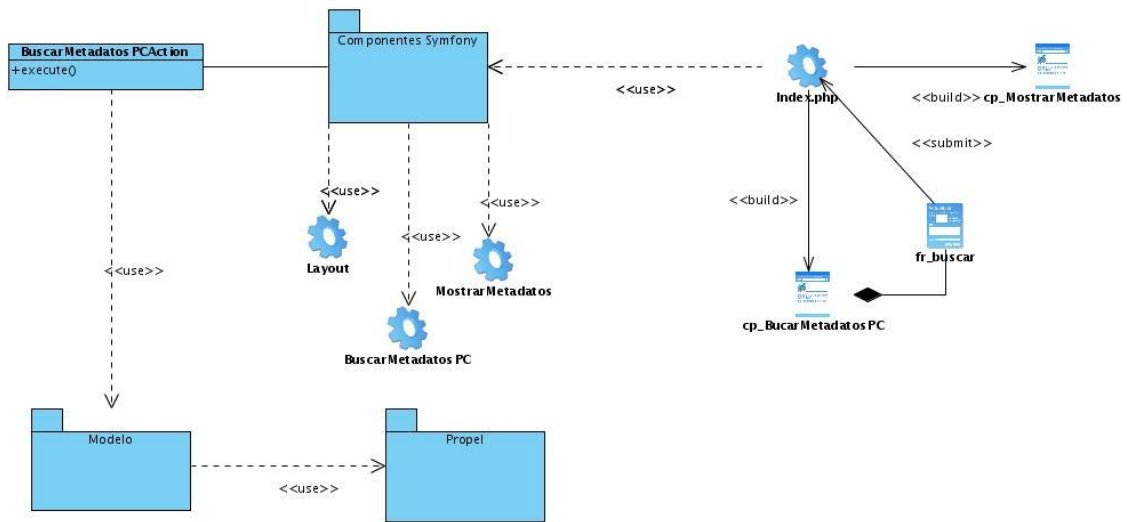


Figura 11: DCD. CU: Consultar Metadatos por Palabras Claves.

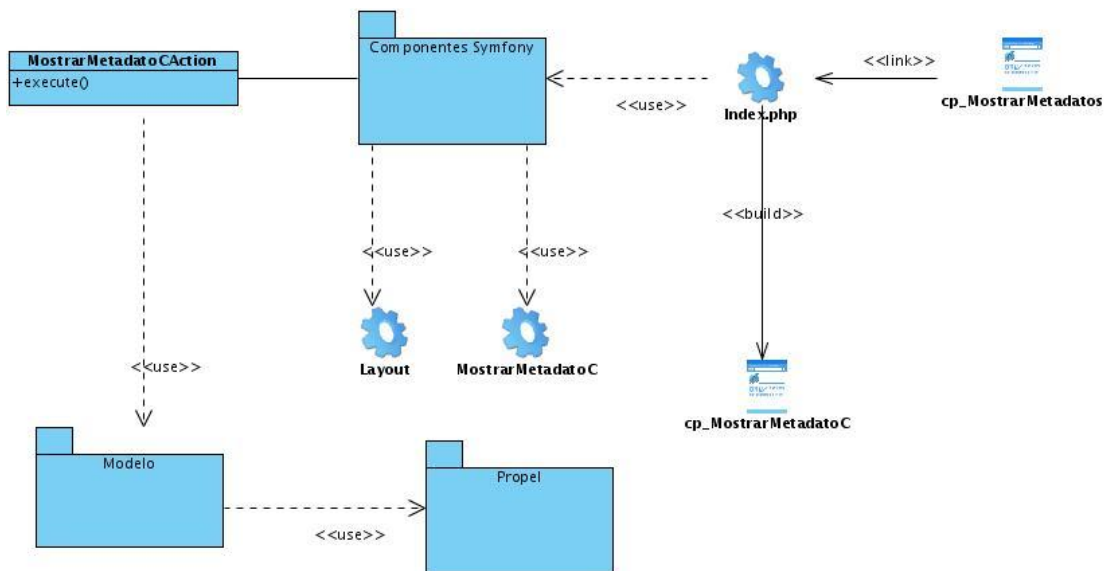


Figura 12: DCD. CU: Mostrar el Metadatos Completo.

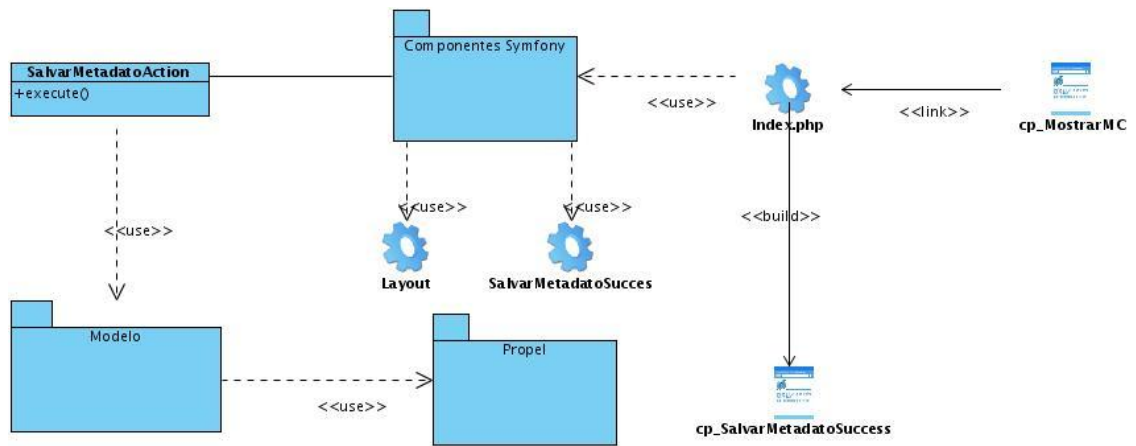


Figura 13: DCD. CU: Salvar Metadatos.

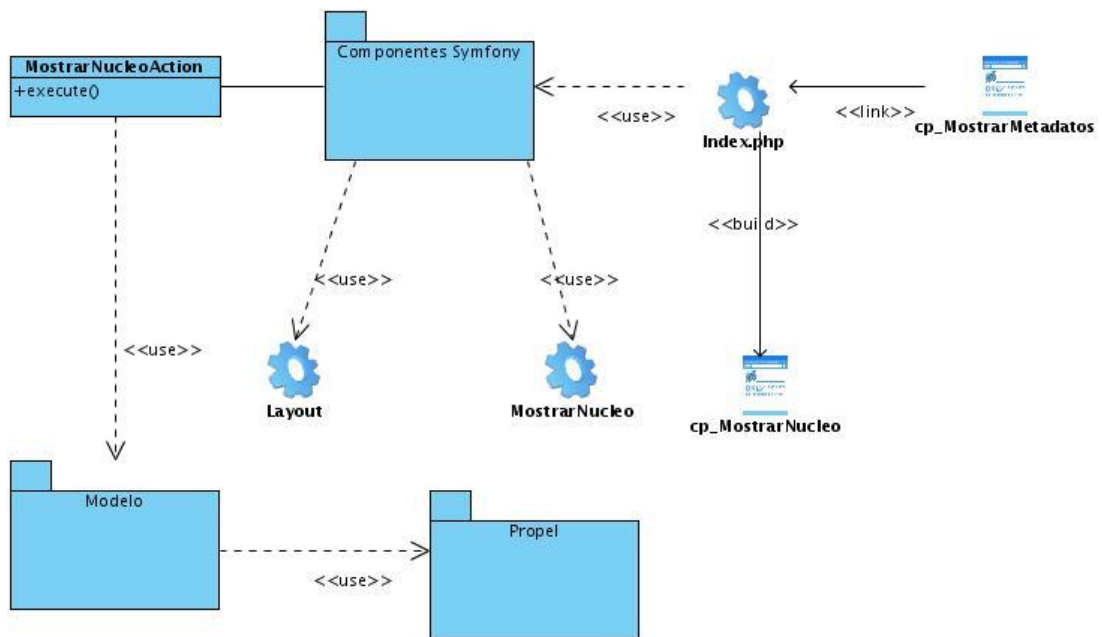


Figura 14: DCD. CU: Mostrar el Núcleo del Metadato.

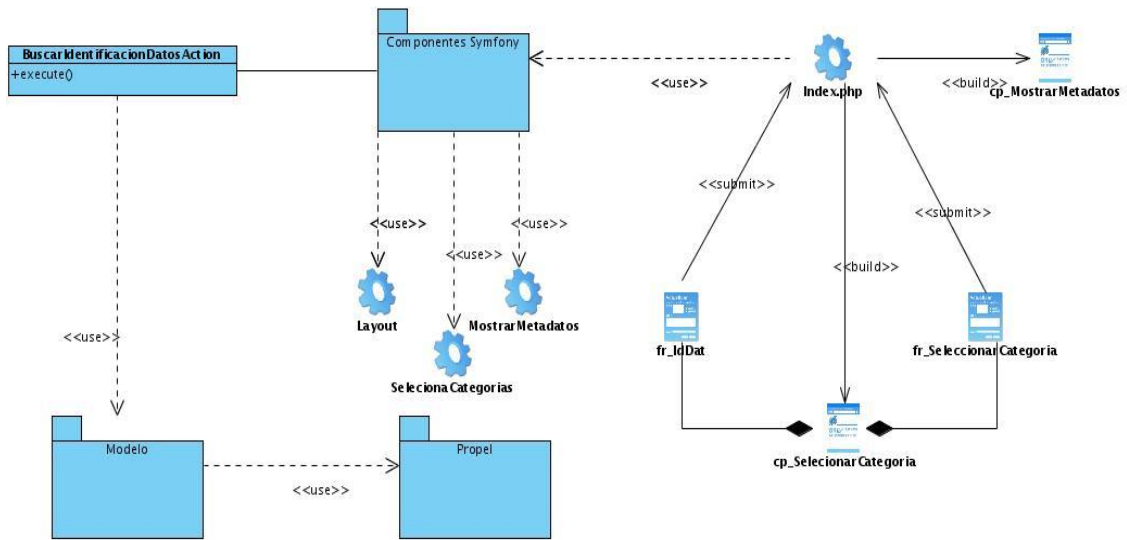


Figura 15: DCD. CU: Consultar Metadatos por Categorías. Sección: Identificación de los Datos

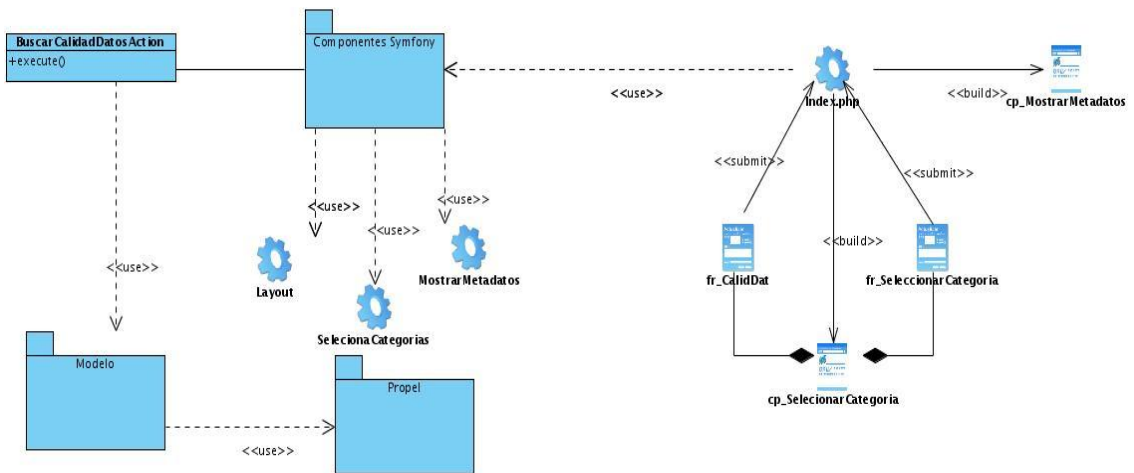


Figura 16: DCD. CU: Consultar Metadatos por Categorías. Sección: Calidad de los Datos.

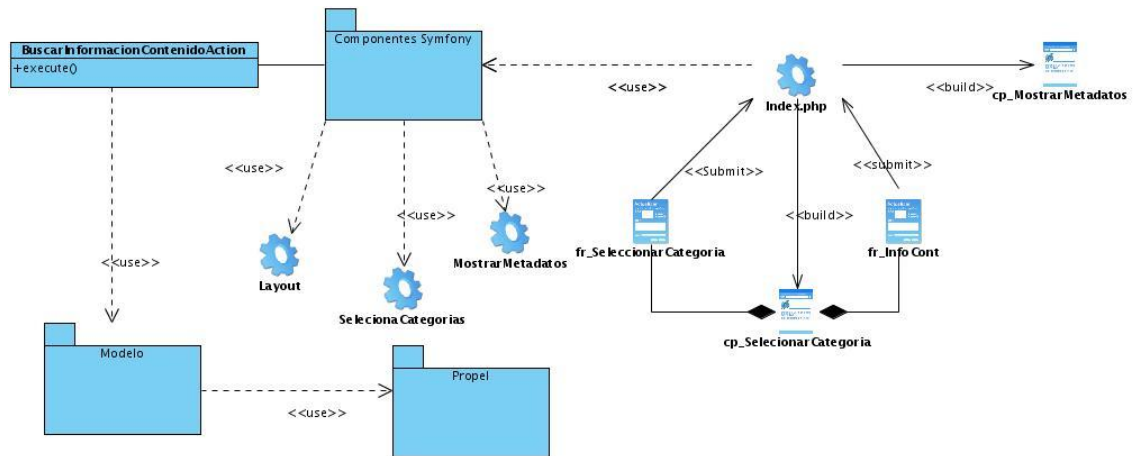


Figura 17: DCD. CU: Consultar Metadatos por Categorías. Sección: Información del Contenido.

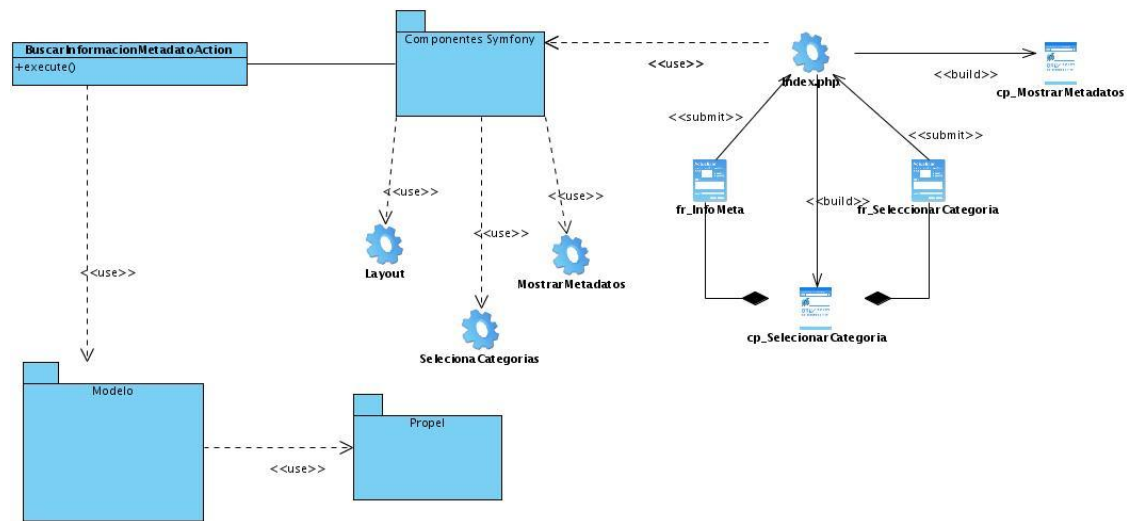


Figura 18: DCD. CU: Consultar Metadatos por Categorías. Sección: Información del Metadato.

#### 4.4 Diseño de la Base de Datos

En el proceso de construcción de todo sistema automatizado, el diseño de la Base de Datos (BD) ocupa un lugar muy importante, a tal punto que éste se puede ver como un proceso independiente dentro del diseño del sistema. Un buen diseño de la BD permite que el acceso a la información sea eficiente (fácil y rápido) y con mínima redundancia.

#### 4.4.1 Diagrama de Clases Persistentes

El modelo de datos de un proyecto Symfony es generado automáticamente por el *framework* y está constituido por cuatro clases por cada tabla de la base de datos, estas son:

baseNombreTablaPeer: Son clases que tienen métodos estáticos para trabajar con las tablas de la base de datos. Proporcionan los medios necesarios para obtener los registros de las tablas y sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada. Representan la parte del modelo que se encarga del acceso a los datos.

baseNombreTabla: Son clases objeto que representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados. Representan la parte del modelo que se encarga de la abstracción de los datos.

nombreTabla, nombreTablaPeer : Sirven para extender las funcionalidades de sus clases bases correspondientes.

La representación genérica del modelo quedaría como se muestra en la siguiente figura:

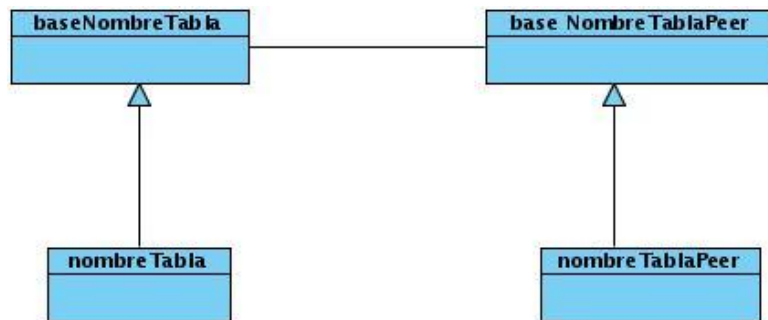


Figura 19: Modelo de Datos genérico en Symfony

Por lo anteriormente planteado y teniendo en cuenta que son las clases `baseNombreTabla` las que mapean las tablas de la base de datos a objetos, se decide en el marco de la investigación, que sean estas consideradas como las clases persistentes.



A continuación se muestra como quedaría la representación de las mismas.

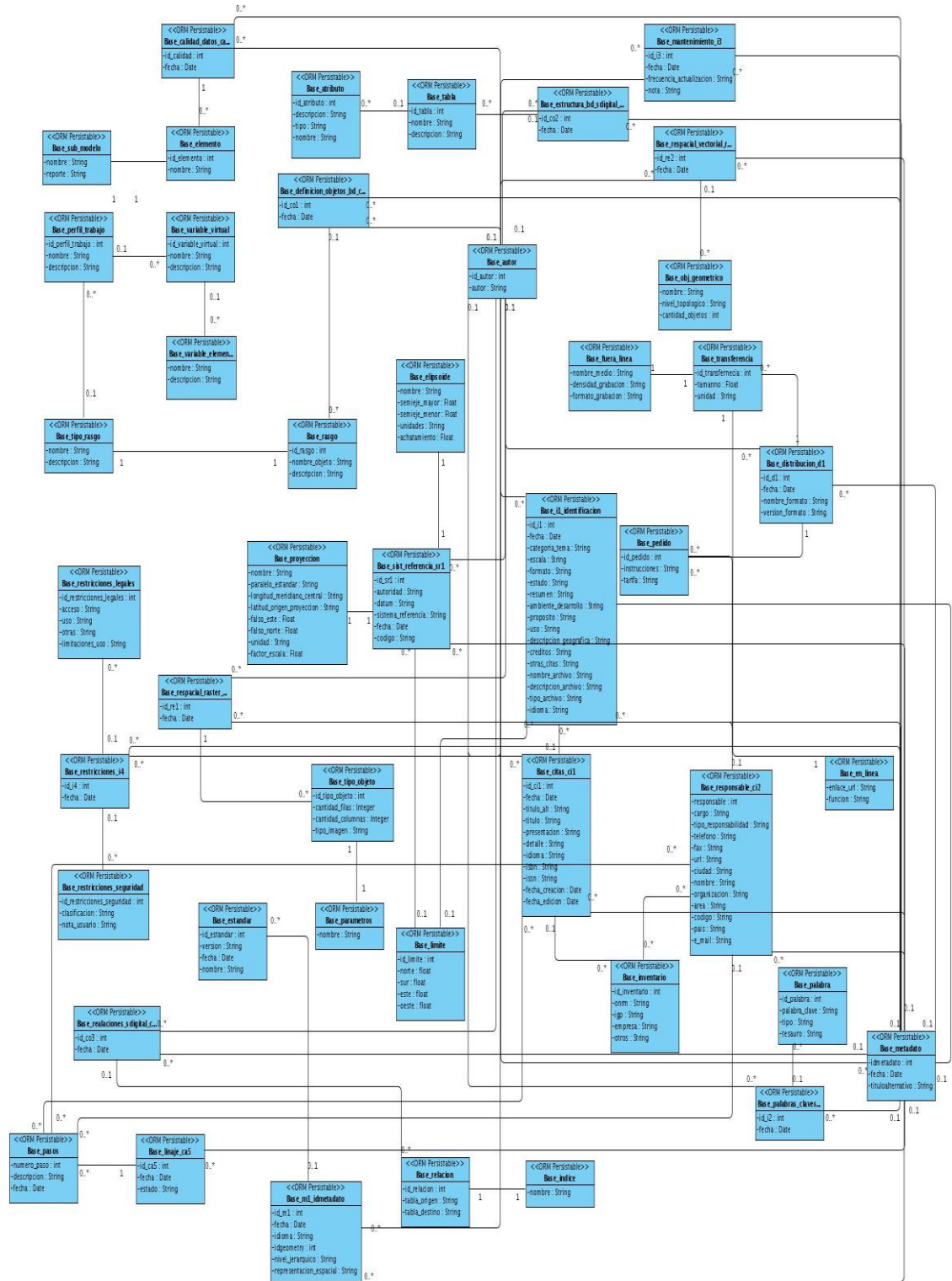


Figura 20: Diagrama de Clases Persistentes.

Las clases del modelo (clases persistentes) son usadas por las acciones (representadas en los diagramas de clases del diseño del epígrafe 4.3.1) para extraer o modificar información en la base de datos. Con el objetivo de ilustrar las relaciones directas que existen entre estas (acciones y clases del modelo), se propone el siguiente mecanismo de diseño.

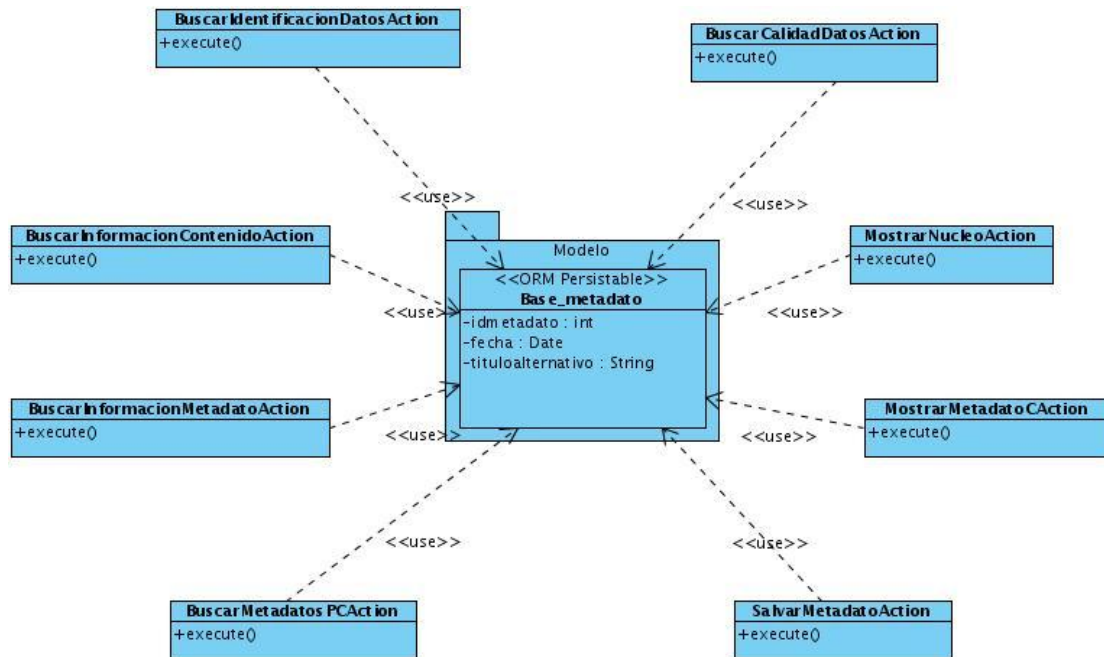


Figura 21: Mecanismo de Diseño

#### 4.4.2 Modelo Entidad – Relación

El Modelo Entidad Relación (MER) está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades, con sus atributos y de las interrelaciones que existen entre estos objetos. Se desarrolló para facilitar el diseño de Bases de Datos permitiendo la especificación de un esquema del universo de discurso, que representa la estructura completa de las mismas. El MER es uno de los diferentes modelos de datos semánticos que existe; el aspecto semántico del modelo reside en su intento de representar el significado de los datos. Este modelo es extremadamente útil para hacer corresponder los significados e interacciones del desarrollo del mundo real con un esquema conceptual (43).

A continuación se presenta el Modelo Entidad – Relación correspondiente al diseño de la Base de Datos de esta investigación:

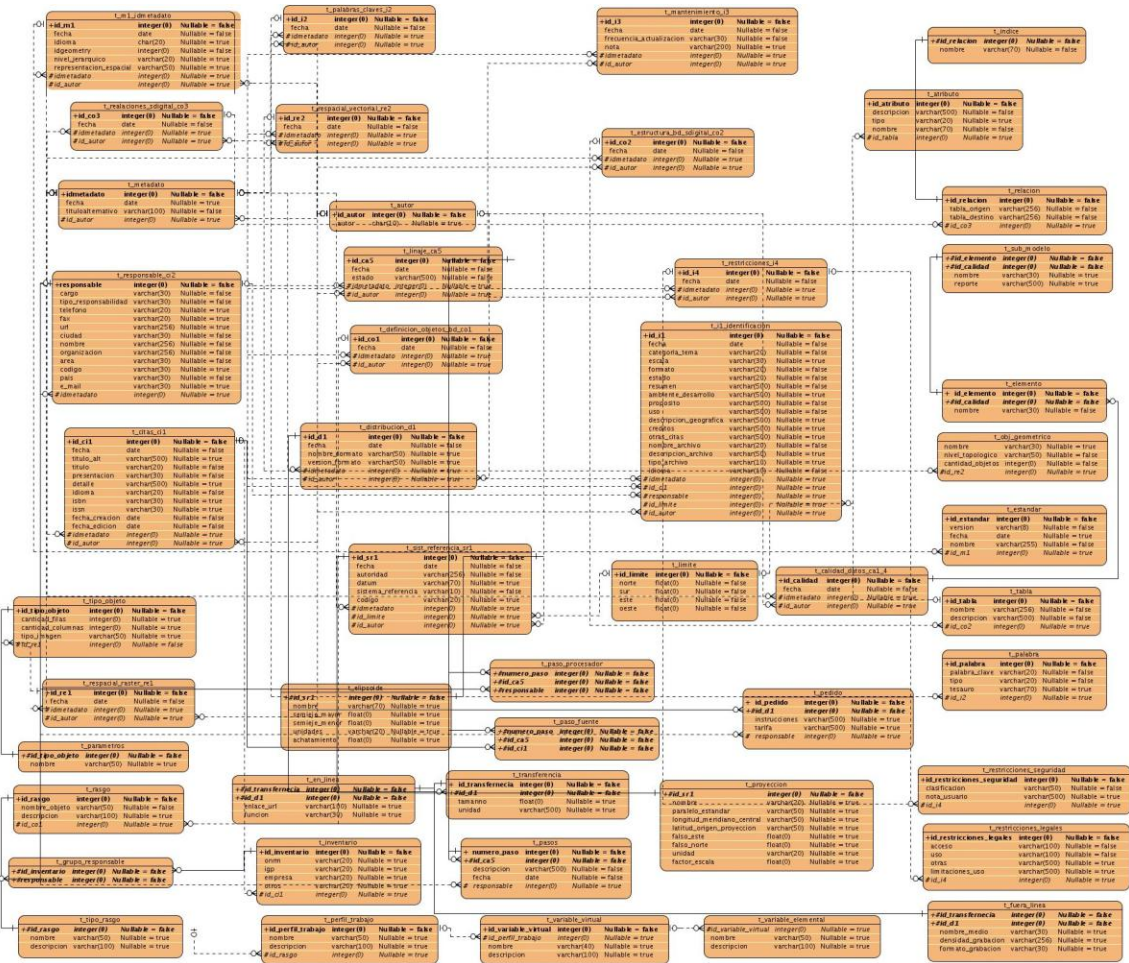


Figura 212: Modelo Entidad - Relación

4.5 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo[27], dando una medida de la tecnología necesaria para el correcto funcionamiento del mismo y de cómo se satisfacen los requerimientos no funcionales de hardware y software.

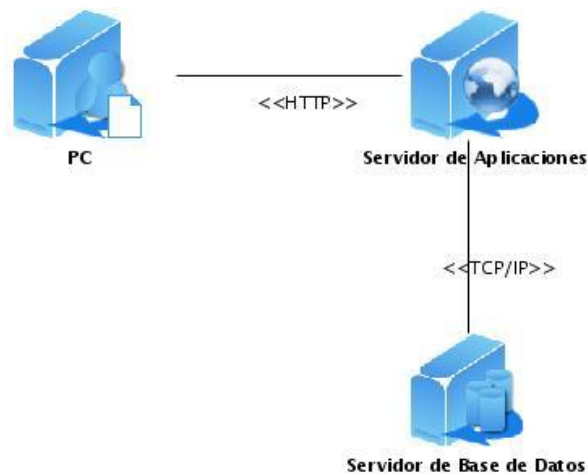
A continuación se muestra el diagrama de despliegue que describe la aplicación que se desarrolla y se describen cada uno de sus componentes.

Nombre del procesador: descripción de la funcionalidad del nodo

PC Cliente: su función es acceder al sistema e interactuar con el mismo según sus necesidades. Podrán acceder tanto internamente como a través de internet. Su conexión es a través del Protocolo de Transferencia de Hipertexto (HTTP por sus siglas en inglés).

Servidor de Aplicaciones: es el nodo que albergará el servidor web, responde a las peticiones de las PCs clientes y realiza las peticiones al servidor de base de datos.

Servidor de Base de Datos: es el encargado de almacenar toda la información generada del sistema. Estará recibiendo constantes peticiones del Servidor de Aplicaciones, con el que se comunicara mediante TCP/IP.



*Figura 223: Modelo de Despliegue*

#### **4.6 Modelo de implementación**

El modelo de implementación describe cómo los elementos del diseño, se implementan en términos de componentes. Cómo se organizan estos de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y cómo dependen uno de otros.[42]

Para estructurar el modelo de implementación se utilizan los diagramas de componentes que muestran las dependencias entre los elementos de implementación y los correspondientes elementos del diseño que son implementados.

La representación gráfica de los diagramas de componentes de esta aplicación, dividida por casos de uso en correspondencia con el diseño realizado, sería la siguiente:

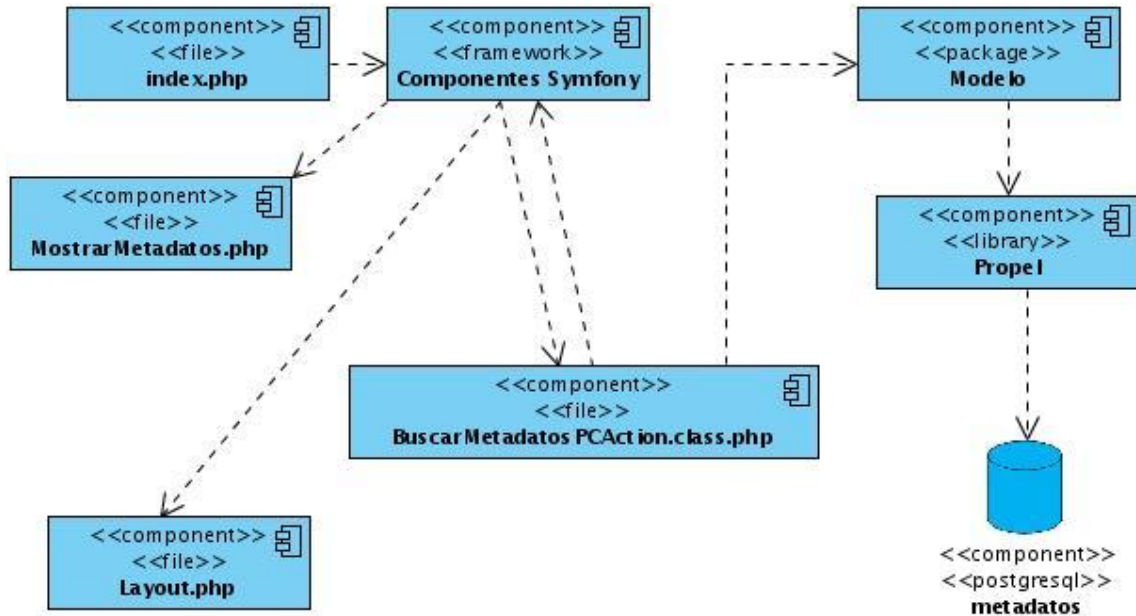


Figura 234: Diagrama de Implementación. CU: Consultar Metadatos por Palabras Claves.

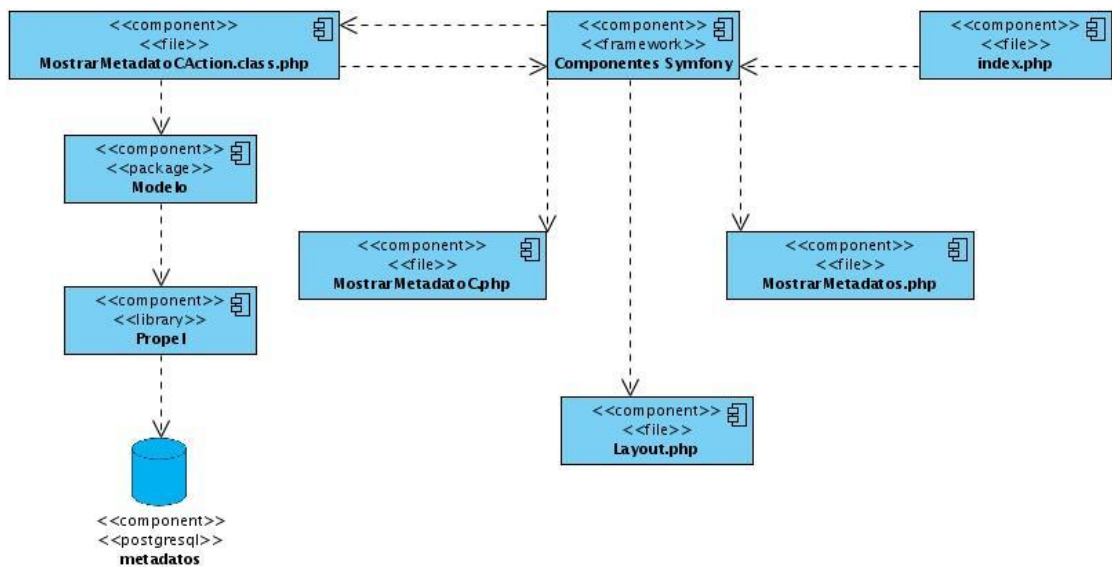


Figura 245: Diagrama de Implementación. CU: Mostrar el Metadato Completo.

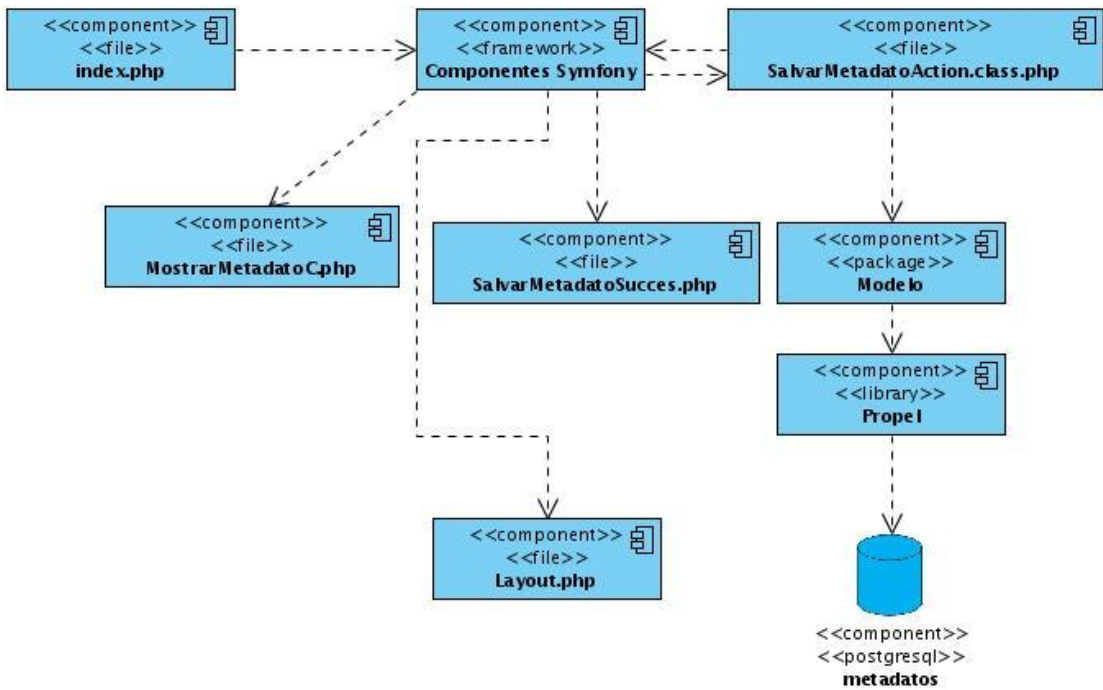


Figura 256: Diagrama de Implementación. CU: Salvar el Metadato.

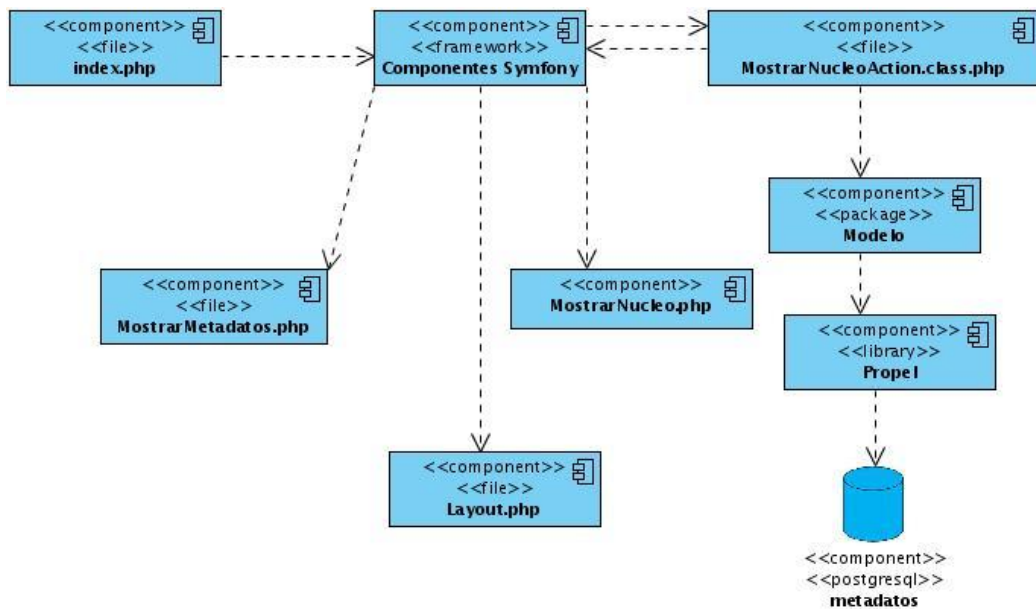


Figura 267: Diagrama de Implementación. CU: Mostrar el Núcleo del Metadato.

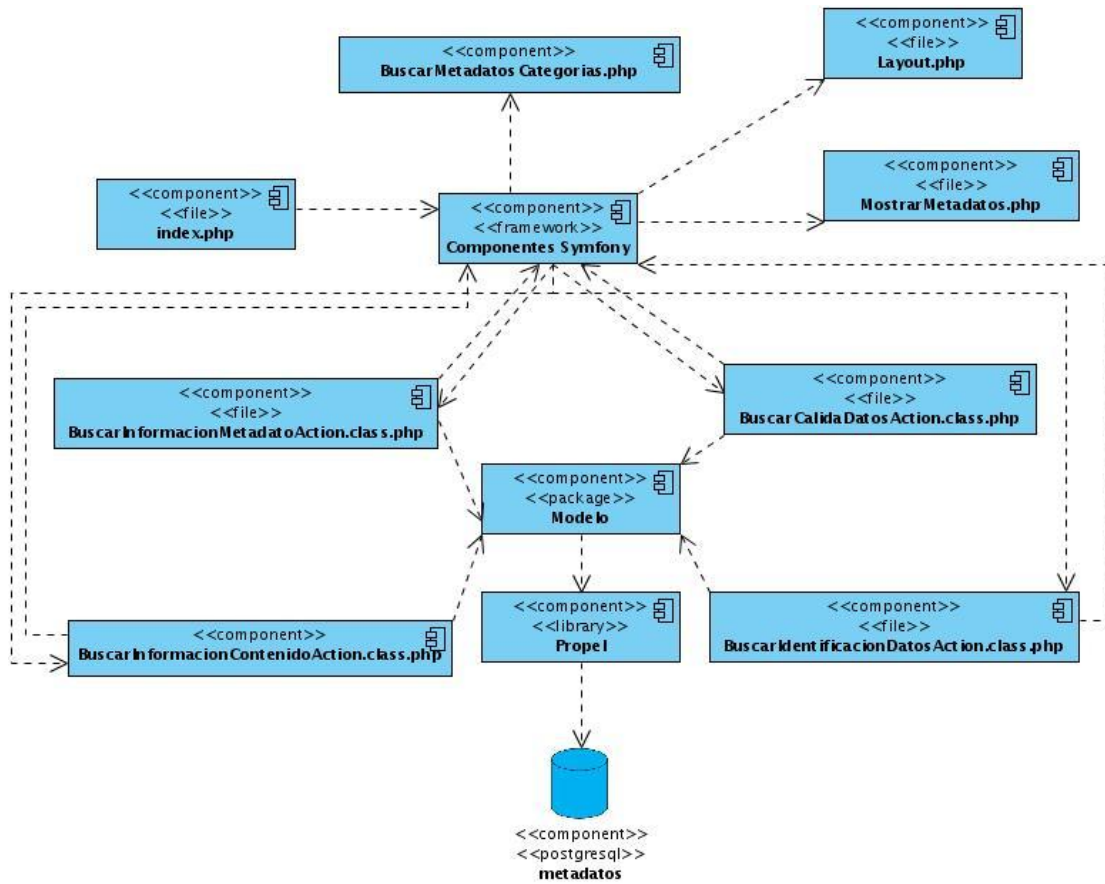


Figura 278: Diagrama de Implementación. CU: Consultar Metadato por Categoría.

#### 4.7 Conclusiones

Con el desarrollo de este capítulo se han obtenido los principales artefactos que se generan durante los flujos de trabajo de Análisis y Diseño (Modelo de Análisis y Diseño) e Implementación (Modelo de Implementación), logrando que de esta manera quede modelada la construcción de la solución propuesta.

## Conclusiones

1. El análisis de los supuestos teóricos acerca del proceso de recuperación de metadatos geológicos y de la informatización de éstos en la Oficina Nacional de Recursos Minerales, permitió conocer algunas deficiencias que existen en la entidad con relación a la recuperación de información, y comprender que el diseño de una aplicación Web para la consulta de los metadatos geológicos, contribuirá a mejorar las mismas.
2. El objetivo elaborado en el diseño de la investigación fue cumplido, logrando modelar el subsistema para la consulta de metadatos geológicos en la Oficina Nacional de Recursos Minerales.
3. El uso de herramientas de software libre propició como ventaja principal, el mínimo de gastos en lo que se refiere al pago de licencias de software, aspecto que favorece a la Oficina Nacional de Recursos Minerales.
4. Se obtuvo la propuesta de diseño de una aplicación web que proporcionará una mejor comprensión del sistema para su implementación, contribuyendo de esta manera a mejorar la recuperación de los metadatos geológicos en la Oficina Nacional de Recursos Minerales.



## Recomendaciones

Una vez concluida la investigación y basándose en las experiencias acumuladas a lo largo del desarrollo de la misma, se proponen las siguientes recomendaciones:

- Concluir con el proceso de desarrollo del Subsistema para la consulta de metadatos geográficos en la ONRM.
  - Implementación.
  - Prueba.
- Seguir el estudio del proceso de recuperación del metadato en la ONRM, con el objetivo de mejorar el sistema y agregar nuevas funcionalidades al mismo.
- Construir del diccionario de términos geológicos Inglés-Español para la correcta interpretación de la ISO 19139.
- Definir el perfil cubano de la ISO 19115 y definir su variante de la ISO 1939.

## Referencias Bibliográficas

- [1]. Diccionario de la Lengua Española de la Real Academia 2001.
- [2]. Ministerio de la Informáticas y Comunicaciones. Disponible en: <http://www.mic.gov.cu/>
- [3]. Portal UCI. Disponible en: <http://www.uci.cu/>
- [4]. Documento de Arquitectura del PICG v 1.2. 2006.
- [5]. Procedimientos para la Documentación del Metadato. Geominera de Oriente 2007.
- [6]. Sánchez, Alejandra; Javier Noguerras y Daniella Ballari: *Normas sobre metadatos (ISO 19115, ISO 19115-2, ISO19932, ISO15836)*.2008.
- [7]. Dempsey, L. y Heery, R.: *Metadata: A Current View of Practices and Issues*. Journal of Documentation. 1998.
- [8]. Méndez. *Sociedad de la información: Metadatos y futuro de la internet en la recuperación de información de calidad*. 2002.
- [9]. Daudinot. *Organización y recuperación de información en internet: Teoría de los metadatos*. 2006.
- [10]. Santos, Oliva Rafael y Quezada, Orozco Eduardo. *Los Metadatos Geográficos: Actualidad y Estándares*. 2006. Disponible en: <http://www.mappinginteractivo.com/>
- [11]. Croft, W.B: *Approaches to intelligent information retrieval. Information processing and management*. 1987.
- [12]. Korfhage, R. R.: *Information Storage and Retrieval*. 1997.
- [13]. Robertson, S.E. y Spark-Jones, K.: *Relevante Weigthing of searh terms*. Journal Documentation. 1977.
- [14]. Asamblea Nacional del Poder Popular. Ley 76. Ley de Minas 1995.
- [15]. Echeverría, Martínez Manuel: *Las Infraestructuras de Datos Espaciales. Experiencias en su implantación*. 2001.
- [16]. Clerigué, Arrieta R.; P. Lorente Echamendi.; S Ruiz Fontano, y C Grasa Sabando: *Integración de Metadatos en un sistema de Información Corporativo: La IDE de Navarra*.
- [17]. Sistema de Información Geológico Minero de Extremadura. Disponible en: <http://sinet3.juntaex.es/sigeo/web/>

- [18]. Javier Alcalá y Guillem Terradas: Geodata. Disponible en: <http://geodata.es/web/>
- [19]. Instituto Geológico y Minero de España. Disponible en: <http://www.igme.es/>
- [20]. Instituto Geológico Minero y Metalúrgico. Disponible en: <http://metadatos.ingemmet.gob.pe/>
- [21]. Australian Spatial Data Directory (ASDD). Disponible en: <http://asdd.ga.gov.au/>
- [22]. Farre: *Rich Internet Applications*. 2005.
- [23]. Vegas, J: *Desarrollo de aplicaciones web*. 2006.
- [24]. Arquitectura cliente/servidor. Diciembre 2003.
- [25]. Catejón, Juan.: *Arquitectura y diseño de sistemas web modernos*.
- [26]. Clements, Paul. *A Survey of Architecture Description Languages. Proceedings of the International Workshop on Software Specification and Design*. 1996.
- [27]. Jacobson Ivar, Grady Booch, James Rumbaugh: *El Proceso Unificado de Desarrollo de Software. Madrid*. 2000.
- [28]. Reynoso, Carlos y Nicolás Kiccillo: *Introducción a la Arquitectura de Software*. 2004.
- [29]. Cultura Informática: Tecnología Cliente/Servidor. Instituto Nacional de Estadística Informática.
- [30]. Disponible en: <http://www.csi.map.es/>
- [31]. Reynoso, Carlos. y Nicolás Kiccillof: *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
- [32]. Ciudad Ricardo, Febe A: *Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos*.
- [33]. González. *Lenguajes del lado del servidor y del lado del cliente*. 2007.
- [34]. Pérez, Javier: *Introducción a AJAX* 2008. Disponible en: [http://www.librosweb.es/ajax/pdf/introduccion\\_ajax.pdf](http://www.librosweb.es/ajax/pdf/introduccion_ajax.pdf)
- [35]. El Portal AJAX para la comunidad hispana. Disponible en: <http://www.ajaxhispano.com/>
- [36]. Gutierrez: *Kumbia: Por qué programar debería ser más fácil*.
- [37]. Mato, Rosa María: *Diseño de Bases de Datos*. 1999.
- [38]. Conferencia Modelamiento del Negocio. Disponible en: <http://teleformacion.uci.cu/>

- [39]. Conferencia Levantamiento de Requisitos Disponible en: <http://teleformacion.uci.cu/>
- [40]. Conferencia Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis. Disponible en: <http://teleformacion.uci.cu>
- [41]. Conferencia Fase de Inicio. Flujo de Análisis y Diseño. Diagramas de interacción. Disponible en: <http://teleformacion.uci.cu>
- [42]. Jacobson Ivar, Grady Booch, James Rumbaugh: *El Proceso Unificado de Desarrollo de Software*.
- [43]. Conferencia de Bases de Datos. Modelo Entidad - Relación. Disponible en: <http://teleformacion.uci.cu/>
- [44]. Zaninotto, Francois y Fabien Potencier: *Symfony, la guía definitiva*. Disponible en: <http://www.librosweb.es/symfony/>

## Glosario de Términos

**IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronics Engineers*), institución americana responsable de la creación de una gran cantidad de estándares en electrónica e informática.

**IBM:** *International Business Machines*, es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

**LAN:** Red de Área Local (*Local Area Network*). Red de ordenadores de reducidas dimensiones.

**WAN:** Red de Área Ancha (*Wide Area Network*). Una red de computadoras de gran tamaño, dispersa por un país o incluso por todo el planeta.

**Plug-in:** Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

**WYSIWYG:** Es el acrónimo de *What You See Is What You Get* (en inglés, "lo que ves es lo que obtienes"), técnica que ofrece la reproducción exacta en pantalla de un texto tal como aparecería después en formato impreso.

**XHTML:** Formato estricto para crear páginas web compatibles con todos los navegadores.

**CSS:** Hoja de Estilo en Cascada (por sus siglas en inglés *Cascade Style Sheet*), estándar de Microsoft para definir los estilos de texto en una página o sitio HTML en Internet.

**DOM:** Modelo de objetos en documentos (por sus siglas en inglés *Document Object Model*). Es una interface independiente de la plataforma y del lenguaje que permite que los programas y scripts tengan acceso dinámicamente y actualicen el contenido, la estructura y estilo de los documentos.

**XML:** Lenguaje de marcado extensible (por sus siglas en inglés *Extensible Markup Language*). Norma para definir nuevos tipos de documentos. Este lenguaje facilita las transacciones en las bases de datos.

**XSLT:** Lenguaje para tratar estructuras de datos en formato XML, la base de la compartición de documentos en Internet.

**JSON:** (*JavaScript Object Notation*), es un formato ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

**XMLHttpRequest:** Objeto que permite realizar comunicaciones con el servidor en segundo plano.

**JavaScript:** Lenguaje para realizar programas que logran efectos especiales en las páginas desarrollado por la compañía Netscape.

**Unix:** Sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

**Linux:** Sistema operativo multiusuario y multitarea basado en UNIX.

**Windows:** Sistema operativo para ordenadores personales comercializado por la empresa Microsoft.

**PhpDocumentor:** Es una herramienta de documentación para php escrita en php.

**Helpers:** Funciones de PHP que devuelven código HTML y que se utilizan en las plantillas.

**URL:** Localizador uniforme del recurso (por sus siglas en inglés *Uniform Resource Locator* ) Es el mecanismo para identificar una ubicación exacta en el internet.

**API:** Interfaz de programación de aplicaciones (*Applications Programming Interface*), una serie de funciones que están disponibles para realizar programas para un cierto entorno.

**Factory:** Patrón de diseño perteneciente a los patrones de diseño GoF.

**Mixin:** Mecanismo de Symfony para extender y modificar las clases existentes en el *framework*.

**Licencia MIT:** Licencia de software que ha empleado el MIT (*Massachusetts Institute of Technology*) Esta licencia permite reutilizar el software así licenciado tanto para ser software libre como para ser software no libre.

**Http:** Abreviación de la designación inglesa para Protocolo de transferencia de hipertexto. Se trata del protocolo más utilizado para transferir datos entre un servidor y otra máquina.

**BeOs:** Sistema operativo tipo UNIX orientado a la multimedia.

**Open source:** Cualidad de algunos softwares de incluir el código fuente en la distribución del programa.

**MAC:** Dirección de hardware de un dispositivo conectado a un medio de red compartido.

**IP:** Ver TCP/IP.

**Rollback's:** Operación que devuelve a la base de datos a algún estado previo

**SDK:** Kit de Desarrollo de Software (por sus siglas en inglés *Software Development Kit*), un conjunto de aplicaciones para desarrollar programas en un determinado lenguaje o para un determinado entorno

**C:** Lenguaje de programación estructurado, de propósito general.

**C++:** Lenguaje de programación orientado a objetos, basado en el lenguaje C.

**Cobol:** Lenguaje de programación enfocado a aplicaciones de gestión.

**PC:** Ordenador personal (*Personal Computer*).

**GNU:** Proyecto para la creación de un sistema operativo de libre distribución.

**GPL:** *General Public License*. Es una licencia de distribución de software gratuito que permite copiarlo, modificarlo y redistribuirlo.

**CUPET:** (Cuba Petróleos) Empresa cubana encargada de la extracción, refinamiento y distribución del petróleo y sus derivados.

**GeoMinsal:** Grupo Empresarial Geominero-Salinero cubano.

**TCP/IP:** Protocolo de control de transmisiones y protocolo de la Internet (por sus siglas en inglés *Transmission Control Protocol/Internet Protocol*). Es el conjunto de Protocolos que definen la comunicación Internet.

**Cliente ligero:** Es una computadora (cliente) en una arquitectura de red cliente-servidor que tiene muy poca o ninguna lógica del programa, por lo tanto depende principalmente del servidor central para las tareas de procesamiento. La palabra ligero se refiere a lo pequeña que es la imagen de arranque, quizá no más grande que la requerida para conectar a la red y arrancar un navegador web.

**SIG:** Sistema de Información Geográfica.

ANEXO I. DIAGRAMAS DE CLASES DEL ANÁLISIS

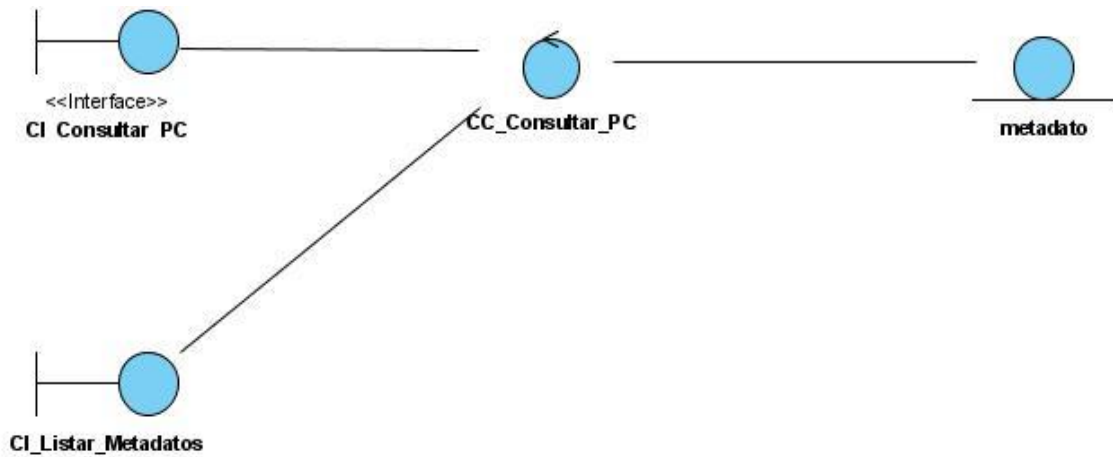


Figura 289: Diagramas de Clases del Análisis. CU: Consultar Metadato por Palabras Claves.

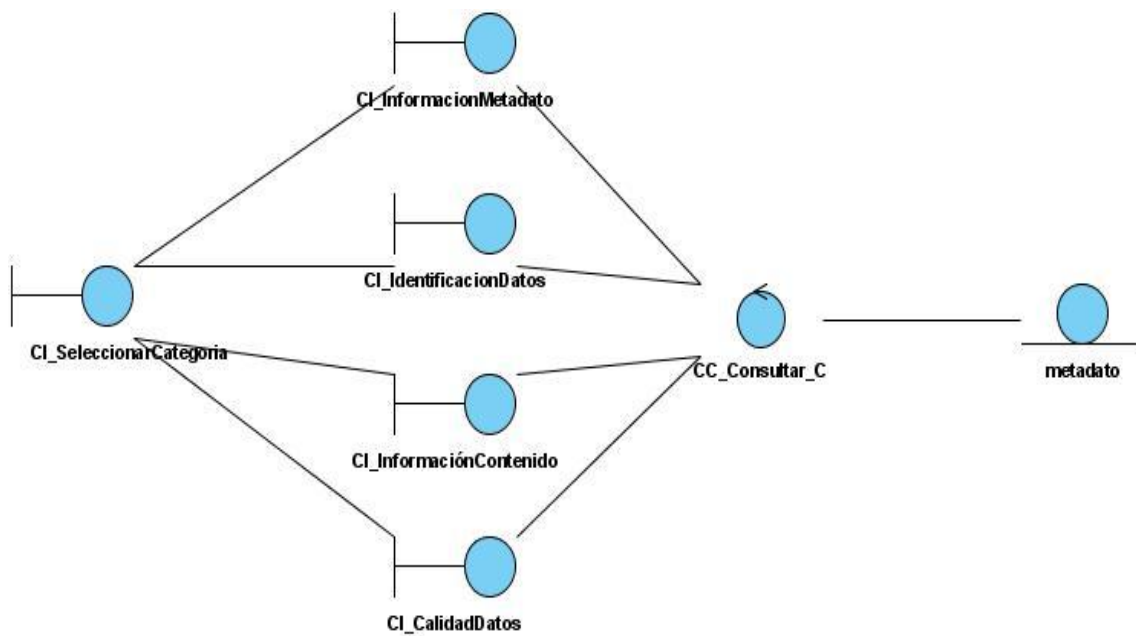


Figura 30: Diagramas de Clases del Análisis. CU: Consultar Metadatos por Categorías



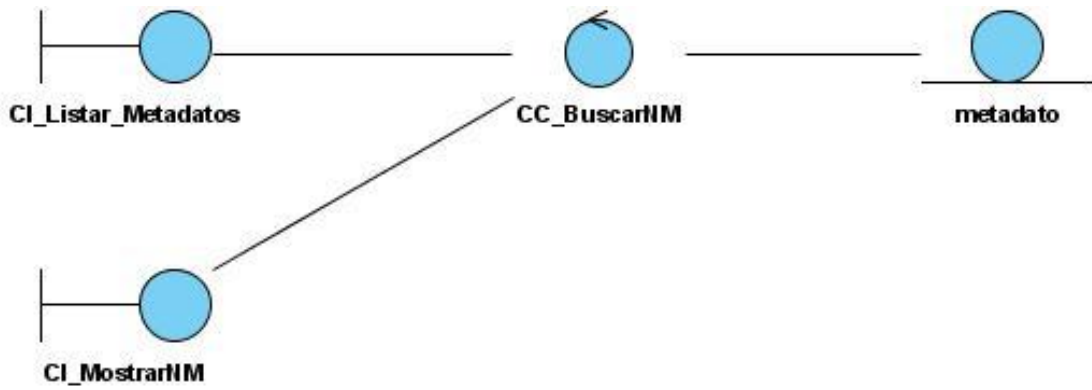


Figura 291: Diagramas de Clases del Análisis. CU: Mostrar Núcleo del Metadato.

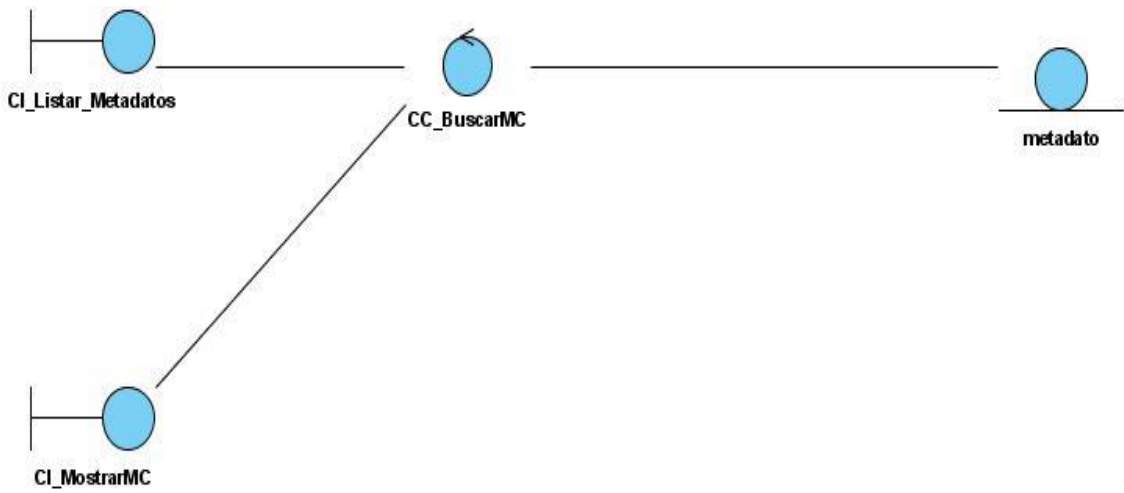


Figura 302: Diagramas de Clases del Análisis. CU: Mostrar Metadato Completo

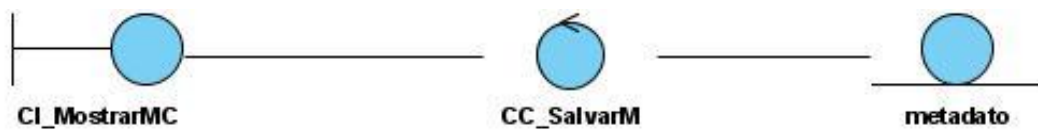


Figura 313: Diagramas de Clases del Análisis. CU: Salvar Metadato.

ANEXO II. DIAGRAMAS DE COLABORACIÓN

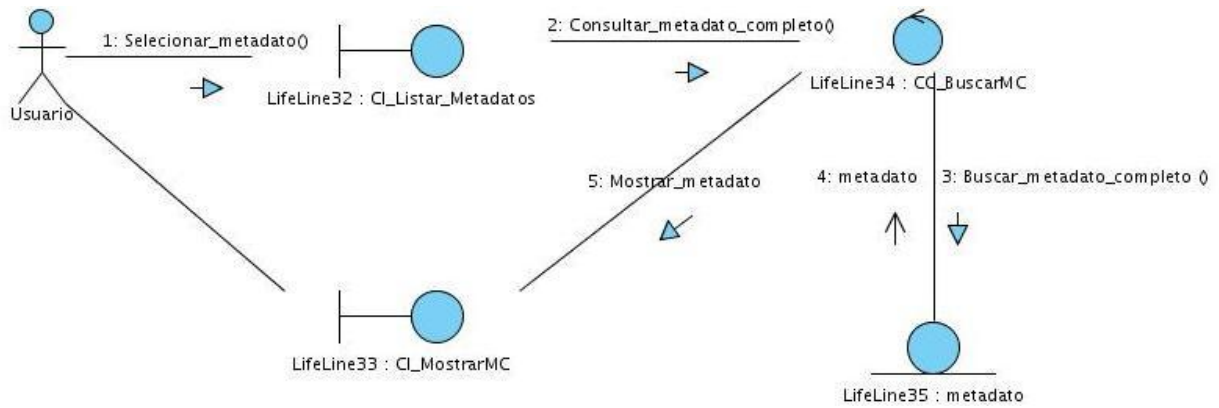


Figura 324: Diagrama de colaboración. CU: Mostrar el Metadato Completo.

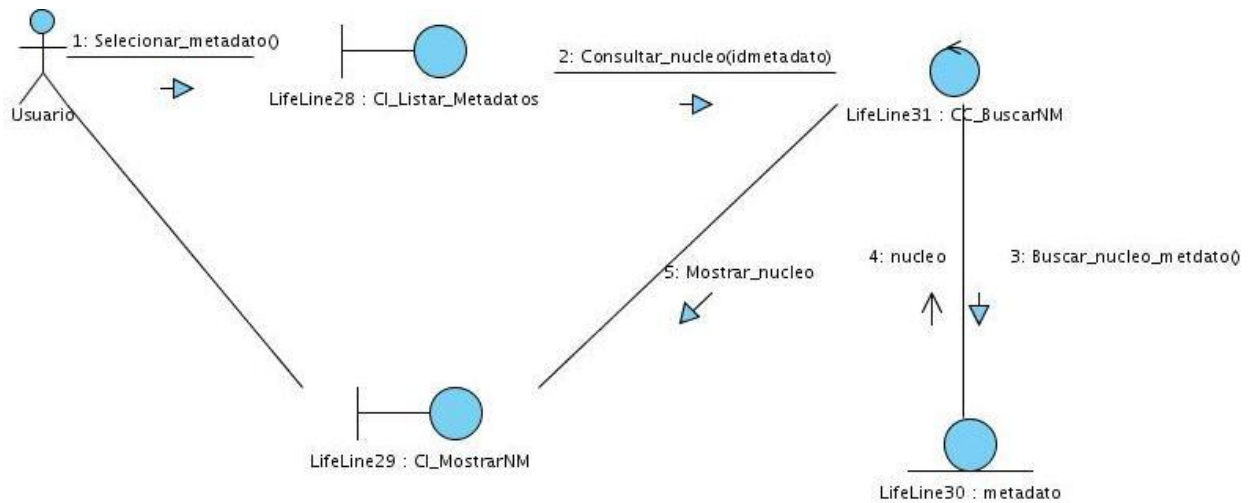


Figura 335: Diagrama de Colaboración. CU: Mostrar el Núcleo del Metadato.

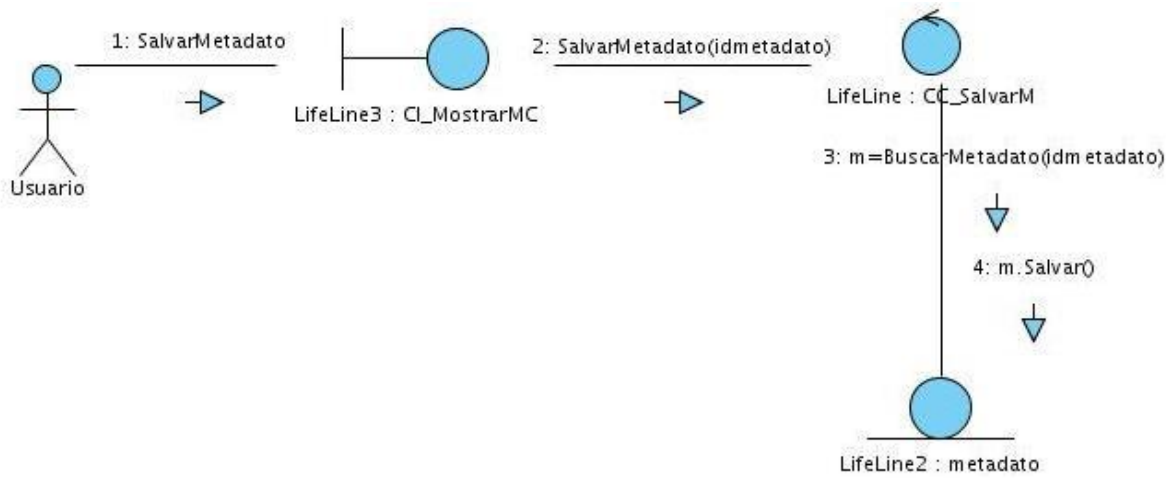


Figura 346: Diagrama Colaboración. CU: Salvar Metadato.

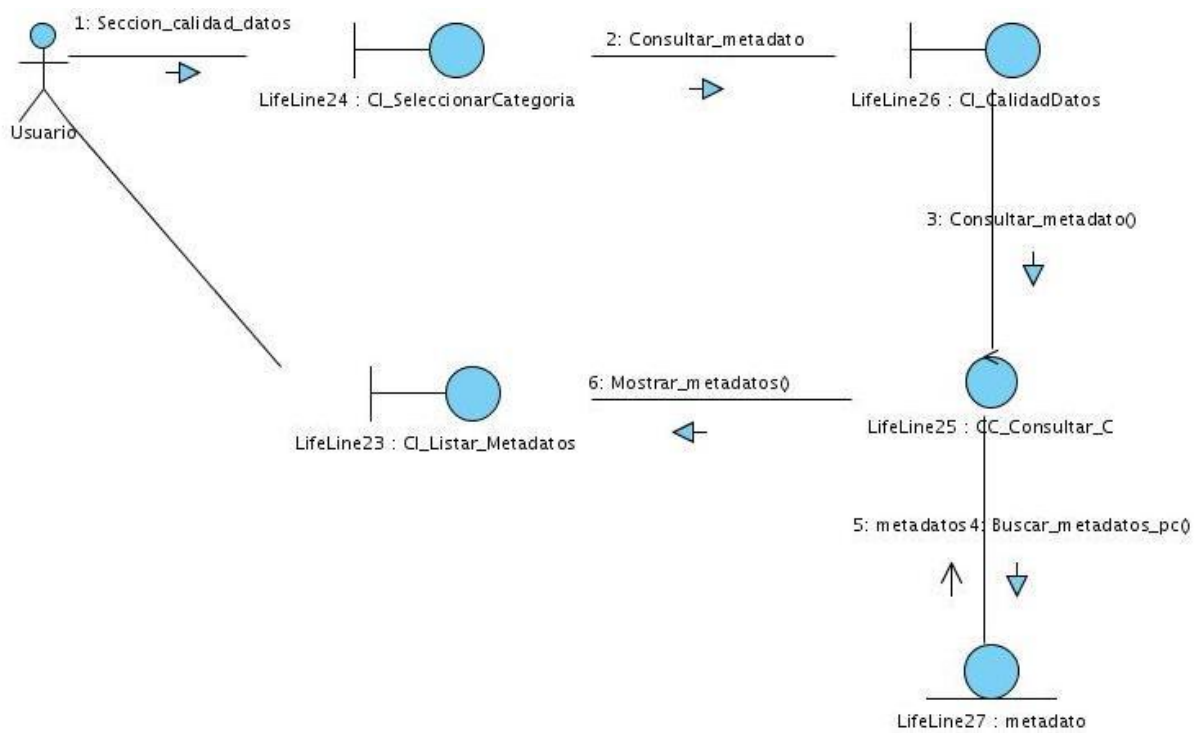


Figura 357: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Calidad de Datos.

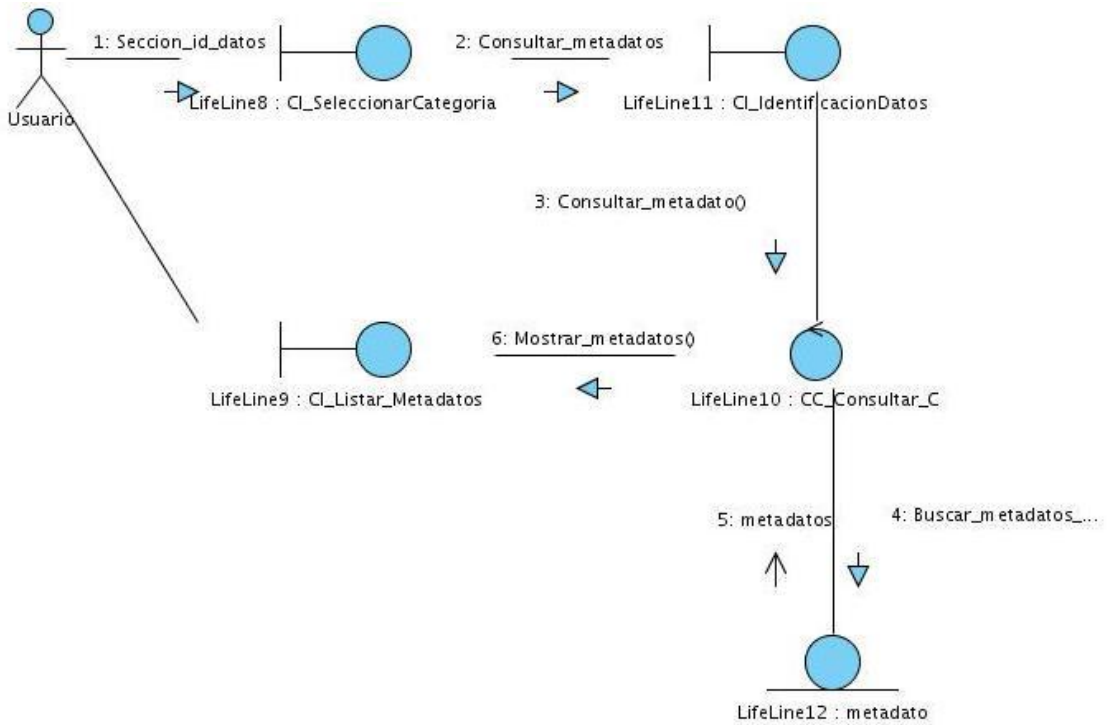


Figura 368: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Identificación de los Datos.

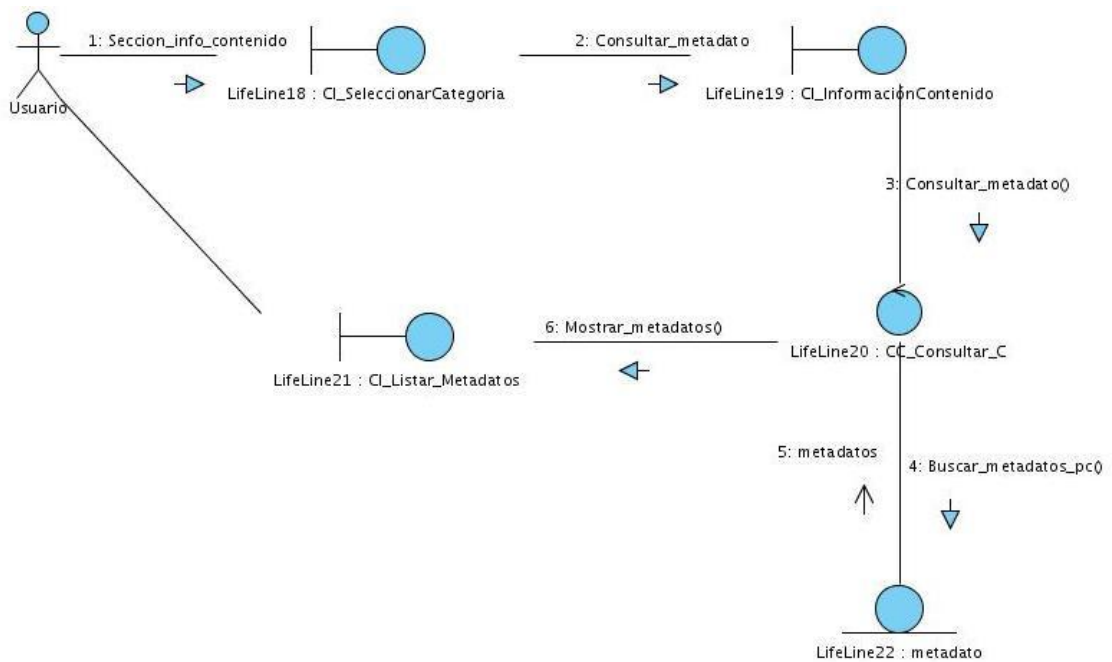


Figura 379: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Información del Contenido.

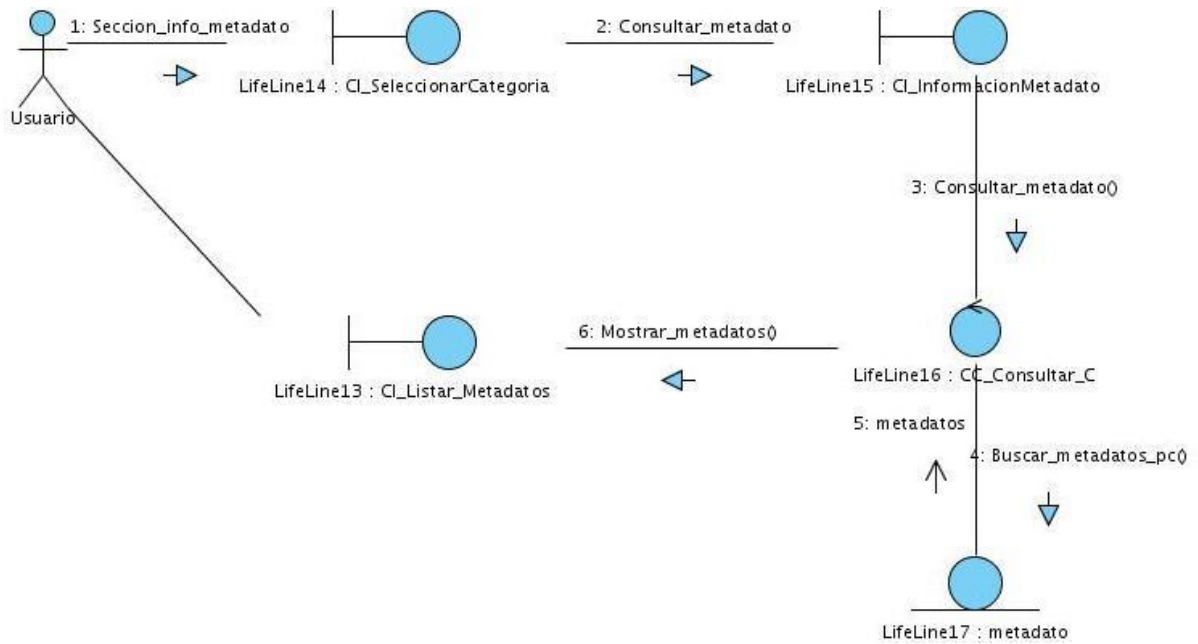


Figura 40: Diagrama Colaboración. CU: Consultar Metadatos por Categorías. Sección Información del Metadato.

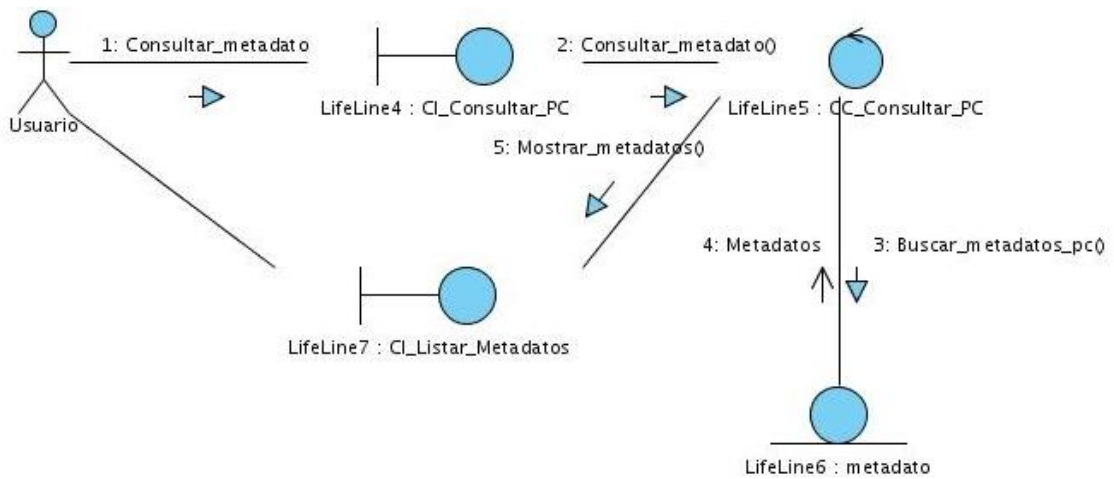


Figura 41: Diagrama Colaboración. CU: Consultar Metadatos por Palabras Claves.

### ANEXO III. ENTREVISTA A ESPECIALISTAS.

Entrevista realizada a los especialistas, Ing. Cesar Rosales y al Ing. Héctor Rodríguez, de manera separada, con el objetivo de conocer a fondo el proceso de recuperación del metadato geológico en la Oficina Nacional de Recursos Minerales (ONRM) así como las funcionalidades con las que deberá contar el sistema a desarrollar.

#### **Datos personales del especialista.**

Nombre:

Años de Experiencia:

#### **Preguntas:**

1. ¿Cómo se realiza actualmente el proceso de recuperación del metadato geológico en la ONRM?
2. ¿Quiénes son los implicados en el proceso?
3. ¿Qué entidades intervienen en el dominio del problema?
4. ¿Qué funcionalidades usted cree que debe tener el sistema?