

**Universidad de las Ciencias Informáticas**

**Facultad 9**



**Título: Análisis y Diseño del Módulo Metadato (Campo).Proyecto PICG.**

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autores:** Ramón González De la Calle  
Alejandro Montejo Rubio

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

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

Julio, 2008.

*“Hay hombres que luchan un día y son buenos. Hay otros que luchan un año y son mejores. Hay quienes luchan muchos años y son muy buenos. Pero hay los que luchan toda la vida: esos son los imprescindibles.”*

*Bertolt Brecht.*

## DECLARACIÓN DE AUTORÍA

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

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

Autores:

Ramón González De la Calle

Alejandro Montejo Rubio

---

---

Tutor: Osmel de la Cruz Amador

---

## DATOS DE CONTACTO

"[insertar breve curriculum e información de contacto del tutor]"

"[insertar breve curriculum e información de contacto del asesor]"

"[insertar breve curriculum e información de contacto del consultante]"

## AGRADECIMIENTOS

### *De Alejandro:*

A mis padres, mi abuela y toda mi familia, por todo el apoyo que me han dado durante todos estos años.

A Yudy , por su cariño incondicional y su compañía.

A mis amigos(as), por aceptarme y ayudarme siempre que los(as) necesité.

### *De Ramón:*

A mis padres, por guiarme siempre por el camino correcto.

A mi familia por el apoyo brindado durante todos estos años.

A mi hermana, por ser tan linda.

A todos mis amistades por compartir conmigo todos estos años, y estar siempre que lo he necesitado.

### *De ambos:*

A nuestros tutores por toda la ayuda prestada.

A todo aquel que hizo posible de una forma u otra la realización del presente trabajo.

A esta Revolución de hombres imprescindibles.

## DEDICATORIA

*A nuestros padres.*

*A nuestra familia.*

*A Yudy.*

*A nuestros amigos y amigas.*

## OPINIONES Y AVALES

## OPINIÓN DEL TUTOR

## RESUMEN

El presente trabajo se desarrolló en la Universidad de la Ciencias Informáticas (UCI), centro en el cual se llevan a cabo, entre muchos otros proyectos, el Proyecto de Nacional de Informatización del Conocimiento Geológico (PNICG), que se desarrolla a petición de la Oficina Nacional de Recursos Minerales (ONRM). El objetivo principal de todo el proyecto es la informatización de los procesos que hoy se realizan en dicha oficina.

Dentro del PNICG, existen varios proyectos en desarrollo entre los cuales se encuentra el Sistema para la Gestión de Metadatos, y este a su vez se divide en tres partes o módulos, Administración, Recuperación y Captura. El presente trabajo se realiza con el objetivo de dar una solución informática a la creación de los metadatos. La investigación se sustenta en una profunda revisión bibliográfica donde se consultó información actualizada sobre el tema, sirviendo como base para desarrollar la Fundamentación Teórica, enfocar metodológicamente el trabajo y seleccionar los métodos, técnicas y herramientas para recoger, procesar y analizar la información necesaria en la creación del sistema.

Se propone como solución una aplicación de escritorio, desarrollada en Java, para la creación de metadatos, almacenando los datos en ficheros XML.

## PALABRAS CLAVE

Metadato, Geología, Aplicación de escritorio, Java, XML.



## DATOS EN INGLÉS

**TITLE:** Metadata Module Automation System (Field Kit). PICG Project.

**Authors:** Ramón González de la Calle.

Alejandro Montejo Rubio.

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

Summary:

This work was developed at Computer Science University (UCI). In this centre it is carried out, among many other projects, the Geological Knowledge Computerization National Project (PNICG), which takes place at request of the Mineral Resources National Office (ONRM). The main objective of this project is the processes computerization that takes place today in that office.

Within the PNICG, there are several projects in development among them the Managing Metadata System and this in turn is divided into three parts or modules, Administration, Recovery and Capture. This work is done with the aim of providing a software solution for the metadata creation. The research is based on a literature thorough review where was consulted updated information about the theme, serving as a basis for developing the theoretical, methodological approach work and selecting methods, techniques and tools to collect process and analyze the information needed in the system creation.

It is proposed settlement as a desktop application, developed in Java, for the metadata creation, storing data in XML files.



## TABLAS Y FIGURAS

Tabla 1	Cartografía Geológica del Territorio Nacional.....	8
Tabla 2	Documentación Geológica en los Archivos de la ONRM.....	8
Tabla 3	Estructura incluidas en GeodatoIC <b>(3)</b> .....	9
Tabla 4	Selección del Programa GeoDato para el perfil de Metadato.....	14
Tabla 5	Definición de actores y trabajadores.....	42
Tabla 6	Descripción textual de los Casos de Uso de Negocio.....	44
Tabla 7	Descripción de los actores.....	45
Tabla 8	Descripción textual del caso de uso: Gestionar Metadato.....	47
Tabla 9	Descripción textual del caso de uso: Actualizar Nomencladores.....	48
Tabla 10	Factor de Peso de los Actores sin ajustar.....	78
Tabla 11	Factor de Peso de los Casos de Uso sin ajustar.....	79
Tabla 12	Factor de Complejidad Técnica.....	80
Tabla 13	Factor Ambiente.....	81
Tabla 14	Esfuerzo del Proyecto.....	82
Figura 1	Paquetes UML de la ISO 19115 para los Metadatos.....	13
Figura 2	Paquetes UML de Metadatos. Relaciones. <b>(5)</b> .....	15
Figura 3	Principales elementos del RUP.....	26
Figura 4	Flujos de trabajo de RUP; <i>términos en inglés</i> .....	26
Figura 5	Vocabulario de UML.....	28
Figura 6	Diagrama de Casos de Uso del Negocio.....	43
Figura 7	Diagrama de Casos de Uso del Sistema.....	46
Figura 8	Proceso guiado por los casos de uso.....	50
Figura 9	Diagrama de Clases: Capturar Metadato.....	51
Figura 10	Diagrama de colaboración: Crear Metadato.....	51
Figura 11	Diagrama de clases: Actualizar Metadato.....	52
Figura 12	Diagrama de Colaboración: Actualizar Metadato.....	52
Figura 13	Diagrama de clases: Actualizar Nomencladores.....	53
Figura 14	Diagrama de colaboración: Actualizar Nomencladores.....	53
Figura 15	Diagrama de clases entidades del paquete: Metadato.....	54
Figura 16	Diagrama de clases del paquete: Información del Metadato.....	55
Figura 17	Diagrama de clases del paquete: Identificación de los Datos.....	56
Figura 18	Diagrama de clases del paquete: Información del Contenido.....	58
Figura 19	Diagrama de clases del paquete: Calidad de los Datos.....	60
Figura 20	Diagrama de clases del paquete: Representación Espacial.....	61
Figura 21	Diagrama de clases de paquete: Sistema de Referencia.....	63
Figura 22	Diagrama de clases del paquete: Distribución de los Datos.....	64
Figura 23	Diagrama de clases del paquete: Información sobre Citas.....	65
Figura 24	Diagrama de clase CU Gestionar Metadato, escenario: Crear Metadato.....	67
Figura 25	Diagrama de clases CU Gestionar Metadato, escenario: Actualizar Metadato.....	68
Figura 26	Diagrama de clases CU: Actualizar Nomencladores.....	69
Figura 27	de componentes: Gestión de Metadatos.....	74
Figura 28	Diagrama de componentes: Metadato.....	75

# ÍNDICE

AGRADECIMIENTOS .....	I
DEDICATORIA.....	II
OPINIONES Y AVALES .....	III
OPINIÓN DEL TUTOR.....	IV
RESUMEN.....	V
DATOS EN INGLÉS .....	VII
TABLAS Y FIGURAS.....	IX
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
<b>1.1 Entrando en la Geología.....</b>	<b>6</b>
<b>1.1.1 Conocimiento Geológico Cubano .....</b>	<b>7</b>
<b>1.1.2 Programa GeoDato<sup>IC</sup> .....</b>	<b>8</b>
<b>1.1.3 Tecnología Base de Datos con Atributos y Registros Variables. BDARV. ....</b>	<b>10</b>
<b>1.2 Metadatos .....</b>	<b>11</b>
<b>1.2.1 Estándar para la Documentación de Metadatos sobre Datos Geológicos. ISO 19115.TC 211. ....</b>	<b>12</b>
<b>1.2.2 Procedimiento para la Documentación del Metadato. Programa Geodato....</b>	<b>13</b>
<b>1.3 Análisis de otras soluciones existentes .....</b>	<b>16</b>
<b>1.4 Pensando en la Solución Práctica.....</b>	<b>19</b>
<b>1.5 Conclusiones.....</b>	<b>20</b>
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR.....	22
<b>2.1 Metodologías de desarrollo.....</b>	<b>22</b>
<b>2.1.1 Metodología RUP (Rational Unified Process). ....</b>	<b>24</b>
<b>2.2 Lenguaje de Modelado UML.....</b>	<b>27</b>
<b>2.3 Sistema Operativo .....</b>	<b>28</b>
<b>2.3.1 Linux.....</b>	<b>28</b>

2.3.2 Ubuntu.....	29
2.4 Lenguajes de Programación .....	29
2.4.1 JAVA.....	29
2.4.2 XML.....	30
2.5.1 NetBeans.....	32
2.5.2 Eclipse.....	33
2.6 Herramienta de Diseño CASE.....	33
2.6.1 Rational Rose.....	33
2.6.2 Visual Paradigm .....	34
2.7 Patrones de Diseño.....	35
2.8 Patrones de Arquitectura .....	37
2.8.1 Modelo Vista Controlador (MVC).....	38
2.8.2 Por Capas.....	38
2.9 Otros Programas Informáticos Utilizados.....	40
2.10 Conclusiones .....	40
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....	41
3.1 Modelo de Negocio.....	41
3.1.1 Actores y trabajadores del negocio.....	42
3.1.2 Procesos de negocio .....	42
3.1.3 Diagrama de Casos de Uso del Negocio .....	43
3.1.4 Descripción textual de los Casos de Uso de Negocio.....	43
3.2 Descripción del Sistema Propuesto.....	44
3.2.1 Descripción de los actores.....	45
3.2.2 Casos de Uso del Sistema.....	45
3.3 Conclusiones.....	49
CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA. ....	50
4.1 Análisis.....	51
4.1.1 Escenario: Gestionar Metadato.- Crear Metadato.....	51
4.1.2 Escenario: Gestionar Metadato.- Actualizar Metadato. ....	52
4.1.3 Escenario: Actualizar Nomencladores.....	53
4.2 Diseño.....	53
4.2.1 Diagrama de clases: Metadato .....	53
4.2.2 Diagrama de clases: Escenario Actualizar Metadato.....	68

4.2.3	<b>Diagrama de clases: Actualizar Nomencladores.</b>	69
4.3.1	<b>Estándares de la interfaz de la aplicación.</b>	71
4.3.2	<b>Concepción general de la ayuda</b>	73
4.4	<b>Modelo de Implementación.</b>	74
4.5	<b>Conclusiones.</b>	76
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD		77
5.1	<b>Estimación basada en el análisis de los Puntos de Casos de Uso.</b>	77
5.1.1	<b>Calcular los Puntos de Caso de Uso sin ajustar (UUCP).</b>	77
5.1.2	<b>Calcular los Puntos de Caso de Uso ajustados (UCP)</b>	79
5.1.3	<b>Estimación del esfuerzo a través de los Puntos de Casos de Uso.</b>	81
5.1.4	<b>Costo del Proyecto.</b>	82
5.2	<b>Beneficios tangibles e intangibles</b>	84
5.4	<b>Análisis de costos y beneficios.</b>	84
5.5	<b>Conclusiones.</b>	84
CONCLUSIONES		85
RECOMENDACIONES		87
REFERENCIAS BIBLIOGRÁFICAS		88
BIBLIOGRAFÍA		89
ANEXOS		91
<b>Anexo 1.</b>	<b>Modelo de Entrevistas con el cliente.</b>	91
<b>Anexo 2.</b>	<b>Diagramas de secuencias de Análisis.</b>	92
<b>Anexo 3.</b>	<b>Diagrama de clases de diseño de interfaz.</b>	94
<b>Anexo 4.</b>	<b>Prototipos de interfaz.</b>	100
<b>Anexo 5.</b>	<b>Paquetes de Implementación del Metadato.</b>	104
<b>Anexo 6:</b>	<b>Paquete de implementación Interfaz Usuario.</b>	108

## INTRODUCCIÓN

Las investigaciones geológicas y mineras en Cuba comenzaron prácticamente desde tiempos de la colonia, manteniéndose durante la pseudo-república aunque no de forma sistemática y con bajo nivel de actividad. Existen reportes de estos trabajos que aún se conservan y constituyeron el embrión o punto de partida de la Geología cubana.

En esta etapa el desarrollo de la minería con la excepción del níquel, fue pobre y se caracterizó por un bajo nivel técnico. La producción minera manejada por los intereses de EEUU se incrementaba durante los conflictos bélicos y bajaban o paralizaban la producción en períodos de paz.

La creación del Servicio Geológico Cubano fue en 1962, con lo cual se comenzó un intenso programa de investigaciones y levantamientos geológicos con la cooperación del entonces campo socialista, manteniéndose esta actividad hasta su desaparición.

En este contexto se realizó el desarrollo de las reservas de níquel y cobalto que sitúa a Cuba entre los principales productores del mundo. A partir de 1990 con la apertura de la economía nacional al capital extranjero, se continuó la exploración de níquel y petróleo con énfasis especial en la exploración de oro, plata, cobre, plomo y zinc.

En 1995 es promulgada la ley 76 “**Ley de Minas**” que regula el proceso concesionario, ordenando y fiscalizando la actividad minera y creando la **Oficina Nacional de Recursos Minerales**, autoridad minero petrolera de la República de Cuba.

Desde la última década del pasado siglo, el país se ha desarrollado bajo condiciones económicas muy difíciles que han determinado la necesidad de buscar nuevas vías que permitan, ante todo, la supervivencia y el desarrollo de nuestra sociedad.

En la actualidad, la economía cubana está muy relacionada con los avances de las nuevas tecnologías de la información y las comunicaciones. Insertarse en ese mercado es uno de los retos principales para la comunidad de profesionales de estas ramas, y con ello contribuir al despegue económico de nuestra nación.

La Industria Cubana de Software (ICSW) tiene la difícil tarea de convertirse en una importante fuente de ingresos, como resultado del correcto aprovechamiento de las ventajas que ofrece el considerable capital humano con el que se dispone.

La Universidad de las Ciencias Informáticas (UCI) y el conjunto de empresas cubanas vinculadas a esta actividad, juegan un papel decisivo en el desarrollo del software en Cuba, y en la materialización de los proyectos asociados al programa cubano de informatización.

El Ministerio de la Industria Básica (MINBAS), y todas sus ramas productivas, forma parte importante en el desarrollo del país. Los administrativos del ministerio han desarrollado un plan para la informatización de todos los procesos productivos. Una de las instituciones pertenecientes al MINBAS es la Oficina Nacional de Recursos Minerales (ONRM), que en conjunto con la UCI han puesto en marcha el Programa Nacional de Informatización del Conocimiento Geológico (PNICG), que abarca todas las actividades principales de la institución: control de concesionarios, captación y consultoría de información primaria geológica, control del balance nacional de recursos minerales sólidos y petrolíferos y gestión de la actividad administrativa de la entidad.

El impacto social del proyecto será sin lugar a dudas notable en las siguientes esferas:

- Preservación del alto grado de conocimiento geológico alcanzado, indiscutible conquista del Socialismo.
- Respuesta eficiente y expedita a las necesidades de información para la toma de decisiones y el desarrollo científico y económico de la sociedad cubana en la minería, agricultura, defensa, medioambiente, construcción, desarrollo vial, hidráulico, salud entre otros.
- Posibilidad de acompañar con herramientas informáticas la transferencia de la experiencia socialista cubana en la administración del conocimiento geológico, medio fundamental de producción.

La Informatización de la rama geológica ha transitado todas las etapas de la informatización de la sociedad cubana desde el uso de mini computadoras hasta el ordenador actual. Lo que condiciona una cultura informática adecuada de los especialistas de las ciencias afines cuyos resultados principales se resumen a continuación.

1. Digitalización de datos de los principales recursos minerales como Níquel, Petróleo, Levantamientos aerogeofísicos nacionales, mapas geológicos a diferentes escalas, más de 100 bases de datos de las principales zonas de interés económico, estructuradas y documentadas.

2. Análisis de contenido con (visión informática) de los datos relacionados con minerales sólidos (Programa **GeoDato**).
3. Desarrollo de la teoría para la normalización de las bases de datos geológicos (Tecnología de **Base de Datos con Atributos y Registros Variables. BDARV**).
4. Desarrollo y asimilación de medios de procesamiento de la información geológica (software de minería, GIS, explotación del hardware CAD, entre otros).

Aunque se ha trabajado mucho en cuanto a informatización, aun existen varios aspectos que todavía no han sido analizados, en nuestro país existe la necesidad de preservar de manera sustentable y poner a disposición de la economía nacional, todo el patrimonio geológico, con la agilidad que exige la dinámica de la toma de decisiones en la actualidad, de donde surge la **Situación Problémica** de este trabajo:

Durante el desarrollo del programa de informatización del todo el conocimiento geológico de nuestro país se espera que el volumen de información geológica supere los 4 terabytes, acumulados en diferentes centros de almacenamiento distribuidos en todo el país. En la ONRM no existe un sistema de Metadatos sobre el cual estén catalogados todos los estudios, documentos, mapas, y bases de datos geológicas que gestione la búsqueda y control de esta información.

De esta manera el programa PNICG de la ONRM exige la incorporación de herramientas informáticas robustas, debidamente protegidas y aseguradas que sigan estándares informáticos, de calidad, legales y normativos de las geociencias, que apoyen la gestión, la comunicación, la información y sirvan de soporte a la toma de decisiones y al establecimiento de mecanismos de control y seguimiento del conocimiento geológico.

Para la creación de metadatos geológicos se sigue la norma ISO-19115, desarrollada por el comité técnico 211 de la ISO, en el 2003. Hoy en día la creación de metadatos en la ONRM se realiza de forma manual, automatizar este trabajo constituye nuestro principal problema a resolver.

Tomando en cuenta lo anteriormente expuesto, se plantea el siguiente **Problema Científico** ¿Cómo diseñar un sistema que sea capaz de crear Metadatos siguiendo las normas ISO-19115 para la documentación de Metadatos?

Definiendo como **Objeto de la Investigación:** El “Conocimiento Geológico”, toda información sobre la estructura y la composición del subsuelo en el territorio nacional y la zona económica de Cuba.

Tratándose como **Objetivo General:** Diseñar un sistema informático robusto, que siga estándares de calidad, legales y normativos de las geociencias, que permita la creación y disponibilidad de los Metadatos.

Como **Objetivo Específico** tenemos:

- Realizar el Análisis y Diseño del módulo de Metadatos Kit Campo.

**Idea a defender:**

Con el diseño de un sistema que automatice la creación de los metadatos geológicos, se sentarán las bases para la implementación de una aplicación que brindará mayor eficiencia al flujo y control de los metadatos, al mismo tiempo que llegarán de manera rápida y oportuna para la toma de decisiones.

**El Campo de Acción es:**

- Proceso de creación de Metadatos en la ONRM.
- Elaboración de procedimientos para la actualización, distribución, análisis de los Metadatos Geológicos.

Como **métodos de investigación científicos** se utilizaron:

**Métodos Teóricos.**

**Histórico lógico:** Posibilita el análisis histórico del proceso de gestión de metadatos geológicos en áreas nacionales e internacionales.

**Análisis y la síntesis:** Se analiza toda la bibliografía, los estándares de metadatos, y la documentación los procesos de gestión de información, y se realiza síntesis de los aspectos de interés.

**Modelación:** Se realiza el análisis y diseño del software, utilizando para ello la metodología RUP, y UML como lenguaje de modelado.

**Métodos Empíricos.**

**Entrevistas:** Se realizan entrevistas a los especialistas de la ONRM, que diseñaron el perfil cubano de metadatos. Se intercambia mediante consultas por correo electrónico constantemente, sobre cambios, dudas y aclaraciones. Ver Anexo 1.

Como **Resultado** de este trabajo se espera disponer del análisis y diseño de un sistema que permita la captura de los datos para la creación y disponibilidad de Metadatos en la ONRM.

Para alcanzar estos objetivos se planteó desarrollar las siguientes **Tareas**:

1. Estudio de diferentes metodologías de desarrollo del software, Lenguajes de Programación, y de los procesos de recuperación y administración de información de metadatos dentro de la ONRM.
2. Investigación en Internet sobre otros productos de software que gestionen negocios similares al que se va a desarrollar.
3. Realización del análisis y diseño del sistema.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Este trabajo se desarrolla sobre todo el conocimiento geológico del territorio nacional cubano, de esta manera, en este capítulo se define ¿qué es la Geología como ciencia?, y algunos de los campos en los que se divide, por otra parte se explica en qué consiste el Conocimiento Geológico Cubano y su volumen de información. Se aborda sobre el programa GeoDato y los resultados que produjo, específicamente sobre la tecnología BDARV. Además se analizan diferentes definiciones sobre ¿que son los Metadatos?, y sus diferentes usos. También se realiza un estado del arte sobre los estándares sobre Metadatos geológicos y las diferentes soluciones informáticas que se le ha dado en varias partes del mundo, incluyendo una solución cubana sobre la cual se desarrolla este trabajo.

### 1.1 Entrando en la Geología

La Geología, (del griego *geo*, tierra, y *logos*, estudio) es una ciencia natural que describe la composición del planeta, su ubicación y actuación en el universo: ¿Cómo se ha formado?, ¿De qué está hecho?, su historia y los cambios que han tenido lugar en él, y sobre él. **(1)**

Esta ciencia trata de interpretar los fenómenos que suceden sobre el planeta Tierra (los terremotos, las mareas, el ciclo del agua, la dinámica de la atmósfera, etc.). Define las leyes que rigen estos eventos e intenta hallar respuesta a los enigmas como son la génesis, la edad de la tierra, la fuente de energía para los movimientos de traslación y rotación, etc. **(2)**

Debe tenerse en cuenta la geología aplicada, que incorpora saberes científicos y técnicos al servicio de las necesidades del hombre. Se aplica, entre otras cosas, a los yacimientos mineros, la explotación del petróleo, el desarrollo del agua, los estudios necesarios para apoyar la construcción de las presas, la generación de energía hidroeléctrica, etc.

La geología se considera dividida hoy en numerosos campos de desigual desarrollo y nivel de especialización. La mineralogía se dedica al estudio y clasificación de los minerales, y una rama de ella, la cristalografía, describe las propiedades geométricas de los cristales que componen la mayoría de los minerales. **(2)**

La petrografía estudia las rocas, que son la forma natural que presentan los minerales combinados y aglutinados. La sedimentología estudia los sedimentos y, a través de ellos, las características del medio en que se formaron, concediendo cada vez mayor importancia al clima. Pero la vertiente más histórica de la geología se fundamenta principalmente en la paleontología y en la estratigrafía. Esta permite deducir la antigüedad de las rocas y sus avatares a través del orden y disposición actual de los estratos, mientras la paleontología basa sus hipótesis de datación y cronologías en el estudio de los restos fósiles acumulados.

**(2)**

La geología estructural, llamada geotectónica, se ocupa de las causas y los mecanismos de los movimientos, deformaciones, pliegues y fallas de las capas. La teoría de la tectónica de placas ha supuesto, en este sentido, una propuesta verdaderamente explicativa e integradora de la llamada geología dinámica, que estudia los procesos que afectan a la corteza terrestre y subdivide sus investigaciones entre la geodinámica interna, que incluye la orogenia, el vulcanismo y la sismología, y la geodinámica externa, dedicada a los procesos morfo genéticos, es decir, la erosión con sus distintas fases de alteración, transporte y sedimentación. **(2)**

### **1.1.1 Conocimiento Geológico Cubano**

La **ONRM** es la encargada de administrar el conocimiento geológico y toda la información geólogo minera petrolera de la República de Cuba.

La expresión tangible de ese conocimiento es el resultado de los estudios de la Geología del país que puede ejemplificarse con las tablas siguientes:

Escala	Área Cubierta (km2)	% de Cuba	Estado
1:500 000	110 992	100	Digital
1:250 000	110 992	100	Digital
1:100 000	110 992	100	Digital
1:50 000	47928	43	No

Tabla 1 Cartografía Geológica del Territorio Nacional.

Expedientes de derechos mineros en el registro minero	450
Informes y Reportes Geológicos	5,389 (1 millón de hojas)
Proyectos Geológicos	1,000
Pozos Petróleo	2,400

Tabla 2 Documentación Geológica en los Archivos de la ONRM.

### 1.1.2 Programa GeoDato<sup>IC</sup>.

A partir de los años 70, se crean diferentes tipos de tecnologías de base de datos para la industria niquelífera desarrollada por la oficina Geominera de Oriente. A finales de los 80 se emprende un proyecto para la creación del sistema de manipulación de Base de Datos Geológicos, pero fue a mediados de los 90 cuando tuvo éxito el desarrollo de un Sistema de Gestión de Bases de Datos Geológicos GEODAT. A partir de este momento se crearon numerosas base de datos geológicos por todo el país, que tenían muy variados formatos en dependencia de la oficina que las desarrollara y los datos que contenían, las que hoy suman

casi un centenar que por su diversidad de formatos y contenidos se hace muy difícil su uso. **(3)**

Es en este contexto donde surge el **Programa GeoDato** y con él la tarea de creación de un sistema que permitiera la carga de los datos primarios de los informes que contienen los archivos del servicio geológico nacional a una base de datos digital.

**Geodato<sup>IC</sup>** es el software que permite la creación y mantenimiento de las Bases de Datos Geológicas cuyo análisis de contenido y modelos lógicos fueron realizados por diferentes grupos de diseño de las empresas de la Unión GeominSal. **(3)**

En la siguiente tabla se muestran las estructuras que fueron utilizadas en la realización de este software:

No.	Estructura	Empresa
1	Puntos de Información Geológica (muestras, puntos, pozos y laboreos)	Geominera Centro
2	Información Primaria de Topografía	Geominera Camaguey
3	Petrografía y Mineralogía.	Instituto de Geología y Paleontología
4	Información primaria Geoquímica.	Geominera Oriente
5	Información Primaria de los Resultados Físicos y Mecánicos.	Geominera Oriente
6	Datos Geológicos y Paleontológicos.	Geominera Centro
7	Geofísica (Magnetometría, Gravimetría y Radiometría)	Instituto de Geología y Paleontología

Tabla 3 Estructura incluidas en GeodatoIC **(3)**

Cada una de estas estructuras tenía sus propias especificidades lo que provocaba que cada una tuviera un programa automatizado para su captación por lo cual habría al final siete sistemas diferentes, esto llevaba a un trabajo de programación y documentación muy grande.

(3)

Es en este punto donde se decide utilizar la tecnología **Base de Datos Relacional con Atributos y Registro de Variables (BDARV)** desarrollada por el Grupo de Servicios Informáticos de la Empresa Geominera Oriente.

### **1.1.3 Tecnología Base de Datos con Atributos y Registros Variables. BDARV.**

Para el mejor entendimiento de esta tecnología se relacionan aquí algunos de los conceptos fundamentales que se manejan en ella, que mas adelante se mencionan, y sobre los cuales se estará trabajando.

**Rasgos:** Es la entidad que representa un grupo de elementos georeferenciados del mundo real que contiene información geológica y tiene una representación geométrica dada, contiene los atributos de ubicación de acuerdo con el tipo geométrico establecidos en la ISO 19107 (Polígono, Línea, Punto, Poli línea o Segmento) y atributos adicionales que constituyen las llamadas variables del rasgo. *Ejemplo: Pozos de Perforación, Puntos de observación, trincheras, etc.*

**Tipo de Rasgo:** Es una entidad perteneciente al grupo de entidades representadas por el Rasgo de la cual heredan sus atributos y pueden agregarse otros que constituyen las variables del Tipo de Rasgo. *Ejemplo: Rasgo: Pozo de perforación, Tipo de Rasgo: Pozo estructural.*

**Variables Virtuales:** Son variables virtuales o simplemente variables, los atributos normalizados de las clases persistentes que definen las entidades diseñadas en los modelos lógicos. *Ejemplo: Fecha de perforado el pozo de perforación, Profundidad de muestreo, etc.*

**Variables Elementales:** Las variables elementales son variables que describen a las variables virtuales a diferencias de las virtuales propiamente dichas que describen rasgos,

tipos de rasgos y perfiles de trabajo. *Ejemplo: La variable virtual profundidad para que quede completamente definida tiene a la variable elemental Unidad de medida.*

**Perfiles de Trabajo:** Es la representación digital dentro del **GeoDato**<sup>1c</sup> de un conjunto de datos en un documento geológico. Contienen el conjunto de variables contenidas en una fuente de información dada dentro de los documentos de soporte físico de los datos primarios (Tablas, informes, registros, etc.). En estos perfiles se recogen las variables por niveles jerárquicos, se establecen sus restricciones, orden de captación y llaves de relación de los niveles jerárquicos. Con ellos puede crearse el formulario de captación en correspondencia con la fuente del dato y asegura la vista de usuario de los datos idéntica a su organización en el soporte original. *Ejemplo: Documentación Geológica del Pozo. (Desde, Hasta, Documentación), Punto de medición geofísica. (DGBouguer, DeltaZ, GanmaTotal)<sup>1</sup>, etc.*

## 1.2 Metadatos

(...) los metadatos, en sí, no suponen algo completamente nuevo dentro del mundo bibliotecario. Según Howe (1993), el término fue acuñado por Jack Myers en la década de los sesenta para describir conjuntos de datos. La primera acepción que se le dio fue la del dato sobre el dato, ya que proporcionaba información mínima necesaria para identificar un recurso. En este mismo trabajo se afirma que puede incluir información descriptiva sobre el contexto, calidad y condición o características del dato (...) **(4)**

Atendiendo a la definición antes mencionada, se puede considerar la catalogación como un proceso de creación de metadatos. Teniendo en cuenta que la mayoría de sistemas metadatos han sido creados no solo por profesionales de la información si no también por informáticos, diseñadores de programas, técnicos de sistemas, etc. El concepto de metadato se utiliza como un término neutral, para alejar posibles perjuicios de personas menos cercanas al mundo bibliotecario, pero si se analiza desde el punto de vista de la información distribuida, metadato, como concepto, aporta más información que el término catalogación, ya que incluye contenido del dato, control, información relevante como materias, precio, condiciones de uso, etc.

---

<sup>1</sup> Variables elementales geológicas.

Ecergovac (1999), por su parte, afirma que un metadato describe los atributos de un recurso, teniendo en cuenta que el recurso puede consistir en un objeto bibliográfico, registros e inventarios archivísticos, objetos geoespaciales, recursos visuales y de museos o implementaciones de software. Aunque puedan presentar diferentes niveles de especificidad o estructura, el objetivo principal es el mismo: describir, identificar, y definir un recurso para recuperar, filtrar, informar sobre condiciones de uso, autenticación y evaluación, preservación e interoperabilidad. (4)

De esta manera se ha decidido el desarrollo de un perfil de Metadatos sobre Datos Geológicos, que organice, describa, ubique, e informe sobre todos los recursos informativos geológicos que se tienen en las oficinas de la ONRM y para ello se decidió el estudio y uso del estándar ISO 19115 para la documentación de Metadatos.

### **1.2.1 Estándar para la Documentación de Metadatos sobre Datos Geológicos. ISO 19115.TC 211.**

Desarrollado en el 2003 por el comité técnico número 211 de la ISO<sup>2</sup> con el fin de homogenizar otras normas sobre metadatos geológicos que se habían realizado con anterioridad, e intentar llegar a un consenso internacional sobre metadatos geoespaciales.

En esta norma internacional los metadatos para datos geográficos se presentan en paquetes UML<sup>3</sup>. Cada paquete contiene una o más entidades, que pueden ser específicas o generales. Las entidades contienen elementos que identifican las unidades discretas de metadatos. Las entidades pueden relacionarse con una o más de las otras entidades, pueden estar agregadas y repetirse tanto como sea necesario para satisfacer las necesidades específicas de cada caso.

Siguiendo este estándar la ONRM ha desarrollado un perfil de Metadatos para el uso nacional. Actualmente está siendo usado para la captación manual de los metadatos en las distintas oficinas, que está registrado en el documento Procedimiento para la Documentación del Metadato del Programa Geodato.

---

<sup>2</sup> Organismo Internacional de Normalización.

<sup>3</sup> United Modeling Language (Lenguaje Unido de Modelado).

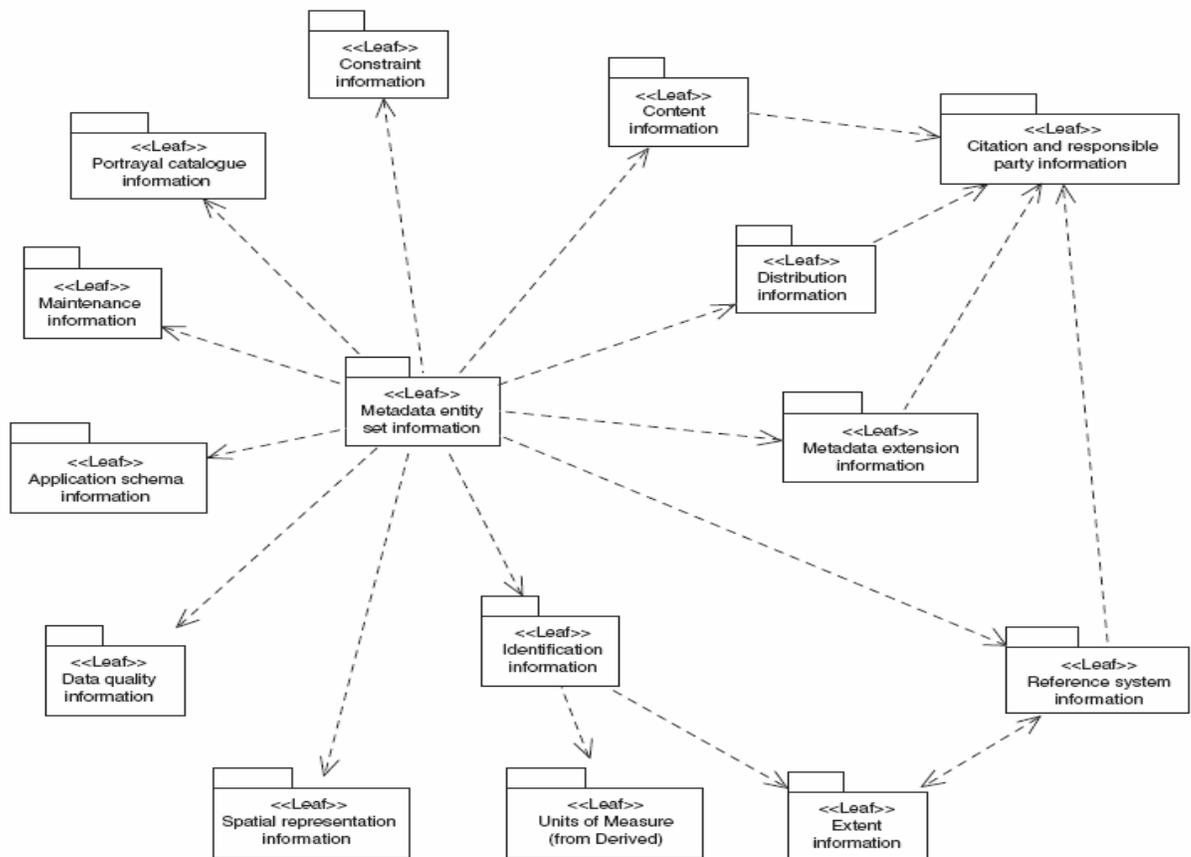


Figura 1 Paquetes UML de la ISO 19115 para los Metadatos.

### 1.2.2 Procedimiento para la Documentación del Metadato. Programa Geodato.

En el marco del programa GeoDato se realizó el análisis de contenido del metadato para los conjuntos de datos geológicos. Siguiendo la norma ISO 19115.TC211, se consideró como conjunto de datos geológico a la unidad de información que se ha digitalizado u organizado como producto de la ejecución de un proyecto de conformación de base de datos. **(5)**

Este perfil de Metadato ha sido conformado escogiendo a ocho de los catorce paquetes UML que propone la ISO 19115, como se relaciona a continuación:

<b>No.</b>	<b>Selección de GeoDato</b>	<b>Paquetes ISO19115/TC211</b>
1	x	Información del Contenido de Entidades Metadato
2	x	Información de Identificación de los datos.
3		Información de las Constricciones
4	x	Información sobre Calidad de los Datos
5		Información sobre Mantenimiento.
6	x	Información sobre la Representación espacial.
7	x	Información sobre el Sistema de Referencias.
8	x	Información del Contenido.
9		Información del Catalogo de Representación
10	x	Información de Distribución.
11		Información de extensiones de Metadatos
12		Información del Modelo de Aplicación
13		Información sobre la Extensión.
14	x	Información sobre Citas y sus Responsables.

Tabla 4 Selección del Programa GeoDato para el perfil de Metadato.

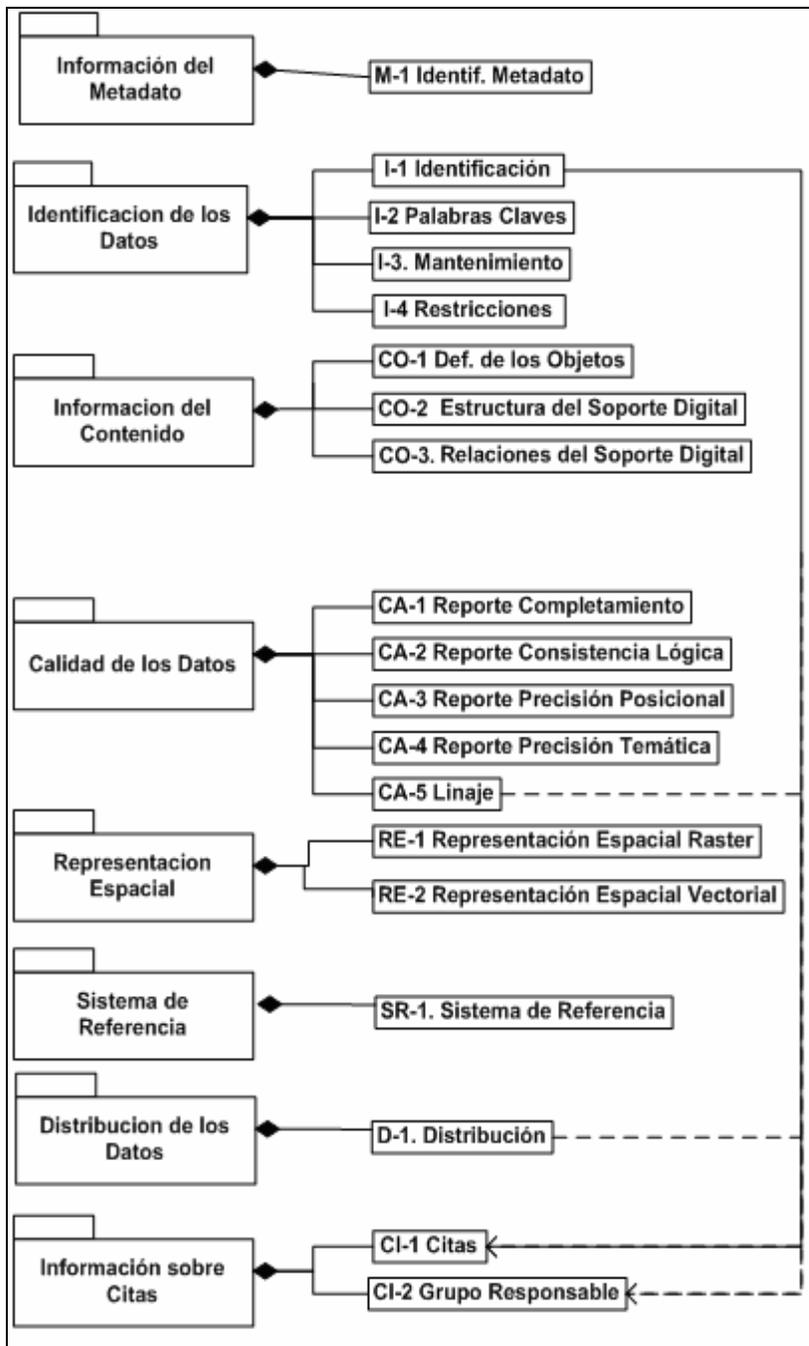


Figura 2 Paquetes UML de Metadatos. Relaciones. (5)

El Metadato quedó formado por un grupo de ocho paquetes relacionados entre sí, que están formados por uno o varios formularios que recogen toda la información.

A continuación se relacionan:

**Información del Metadato:** Establece la información básica acerca de la documentación del metadato para el conjunto de datos.

**Identificación de los Datos:** Recoge la información básica necesaria para identificar el conjunto de datos.

**Información del Contenido:** Para implementar los descriptores de la información geológica, se ha aplicado la tecnología de Base de Datos de Atributos y Registros Variables (BDARV), en esta sección se informa sobre la estructura e información que contiene cada base de datos.

**Calidad de los Datos:** Esta sección recoge el reporte que refiere la calidad de los datos así como las pruebas aplicadas para comprobarla y el resultado obtenido.

**Representación Espacial:** Describe los mecanismos empleados para representar espacialmente los datos contenidos en el conjunto de datos (Ráster o Vectorial).

**Sistema de Referencia:** Describe el sistema de referencia espacial en el cual están definidas las coordenadas de los datos presentes en el conjunto de datos.

**Distribución de los Datos:** En esta sección del metadato se recoge la información acerca del organismo encargado de la distribución de los datos y las condiciones para realizar ese servicio.

**Información sobre Citas:** Toda la información sobre las citas y sus grupos responsables.

Cada uno de estos pequeños paquete de metadatos, sus formularios, y variables han sido muy bien definidos y documentados en el perfil Metadatos. (Para más información consultar el documento “Procedimiento para la documentación del Metadato”).

### **1.3 Análisis de otras soluciones existentes**

En la investigación realizada se ha estudiado algunas otras soluciones que se han dado a problemas similares por todo el mundo. Muchos son los perfiles desarrollados por todas partes del mundo, cada uno, con sus específicas particularidades, adaptando siempre las

características de sus sistemas de datos geoespaciales al estándar ISO para la creación de metadatos.

En Cataluña, España, se ha desarrollado un nuevo perfil llamado Infraestructura de Datos Espaciales de Cataluña (IDEC), en el que analizan muy bien la ISO-19115, y adaptan correctamente sus necesidades específicas, que en comparación con el perfil cubano, sólo añaden un nuevo paquete llamado Información del Distribuidor<sup>4</sup> en el que se organizan los datos del distribuidor, características de el proceso de pedido, tarifas, y disponibilidad de los datos.

Para automatizar este perfil, Cataluña desarrolló su aplicación llamada MetaD, con el objetivo de, además de la creación de metadatos, la edición, el mantenimiento y exportación de metadatos de la manera más sencilla posible para los usuarios. En la misma, se ha desarrollado funcionalidades de importación de metadatos desde ficheros XML y base de datos, para su edición y validación y su posterior exportación.

La aplicación cuenta con una interfaz principal en la que se pueden manejar la entrada y edición de los datos, facilitando una fácil navegabilidad mediante un mapa de aplicación, en el que se actualiza y se muestra el estado de los datos en todo momento, una interfaz para la importación de los metadatos desde ficheros XML y base de datos, un visualizador de XML, y una especie de Ayuda llamada Microhelp, en la que especifica con comentarios sobre las definiciones y características de cada variables y campos donde el usuario este trabajando.

De la misma manera en Navarra, España, se desarrolla su perfil, Infraestructura de Datos Espaciales de Navarra (IDENA), en la que explota las características del Estándar y resuelve sus problemas específicos de una forma más limitada, recomendando un núcleo más pequeño de datos, evitando un esfuerzo excesivo en la creación de metadatos.

Por su parte el Instituto Geológico Minero y Metalúrgico (INGEMMET) de Lima, Perú, ha desarrollado su perfil para los metadatos geológicos, que se ha basado en un 95 % en el ISO, facilitando así compatibilidad entre los catálogos a nivel nacional en el Perú. En cuanto a la

---

<sup>4</sup> Paquete UML Distributor Information en la ISO 19115.

disposición de los elementos se ha seguido el orden de la norma aunque para ello han suprimido algunos paquetes de la propia norma, para ganar en claridad, y manejabilidad, teniendo en cuenta que muchos de los datos han sido distribuidos por los otros restantes paquetes.

Para la automatización del perfil de INGEMMET se estudiaron dos aplicaciones que se encargan de la gestión de los metadatos: ArcCatalog y CatMDEdit. **(6)**

CatMDEdit tiene una muy buena interoperabilidad en cuanto a los cambios que pueda haber en la ISO 19115 que puedan repercutir en la herramienta, los cuales se resolverían en muy poco tiempo. Esta herramienta aporta tesauros a través de una interfaz amigable para su utilización. Sin embargo, CatDMEdit, resulta una herramienta muy complicada en cuanto a la diversidad de campos que hay para completar.

ArcCatalog, por otro lado, resulta mucho mas sencilla en cuanto a los datos que deben ser llenados, sin embargo fue desarrollada en una versión desfasada de la norma ISO 19115, lo que significaría que los metadatos creados no serian entendidos por otras herramientas con la versión actualizada del estándar.

Después del análisis de ambas aplicaciones, INGEMMET optó por usar CatMDEdit, aunque en combinación con ArcCatalog, la cual realizaría la captura de los datos y luego CatMDEdit completaría, y actualizaría más fácilmente los metadatos, creado así una pasarela entre ambas aplicaciones. **(6)**

Todas estas soluciones son válidas, teniendo en cuenta que cada una de las instituciones tiene sus propias características, y el desarrollo de un perfil particular del Estándar. El perfil cubano creado por el Programa GeoDato, satisface todas las exigencias del estándar ISO 19115, adecuándose muy correctamente a las necesidades de nuestro país.

Cuba es un país pequeño y tiene centralizada toda la información geológica, que es localizada y administrada desde la ONRM. Esta oficina desea implementar un sistema de gestión de metadatos, para su publicación e intercambios en Internet, de manera que se hace necesario

seguir todos los estándares actualizados en estos temas, para evitar errores de comprensión por parte de otros sistemas internacionales, tal como le sucede a ArcCatalog.

Por otra parte, es la primera vez que se habla de metadatos geológicos en Cuba, para muchos especialistas será un término nuevo, y su entendimiento deberá estar previsto a partir de una correcta documentación, por tanto, se hace necesario que un sistema informático de metadatos en Cuba ofrezca ayuda, que mantenga orientado al especialista que interactúa con él acerca de todos los campos y variables, y su significado.

Este sistema informático, deberá ser lo más amigable posible, en cuanto a interfaz de usuarios, navegable y comprensible.

#### **1.4 Pensando en la Solución Práctica.**

Teniendo en cuenta los objetivos principales del estándar ISO 19115 y el perfil de Metadatos cubano:

1. Suministrar un conjunto común de terminología y definiciones para documentar datos geoespaciales digitales.
2. Establecer los nombres de los elementos simples y compuestos que han de ser usados, su definición así como información de los valores que deberán ser provistos para estos elementos.
3. Definir estructura, modelo y esquema para describir toda la información geográfica análoga y digital.
4. Permitir la búsqueda, intercambio e interoperabilidad de la información a través de catálogos estandarizados.

El perfil, desarrollado por la Empresa Geominera de Oriente para Cuba, necesita de un sistema que cubra todas las necesidades de la gestión de los metadatos en el país. Durante la concesión del proyecto, el estudio de las funcionalidades necesarias y las particularidades del negocio de gestión de metadatos geográficos en Cuba, se decidió la implementación de tres

pequeños módulos o kits informáticos, para resolver la automatización de la gestión de metadatos en Cuba. Que sigan además los estándares ISO 19115 y el ISO 19139, que controla los esquemas XML para los ficheros de metadatos geológicos y provee además de un diccionario de términos sobre los datos de cada elemento del metadato y toda su información.

Los módulos o kits son:

1. Para la Administración de los datos.
2. Para la Recuperación de los metadatos.
3. Para la Creación de los Metadatos en el terreno.

Este trabajo se centra en la implementación de una aplicación para la creación de los metadatos en el terreno (dígase en todas las instancias de ONRM a lo largo del país), que permita la captura y edición de los datos siguiendo el perfil de los Metadatos cubano, y brinde una fácil operatividad, navegabilidad, mantenimiento y entendimiento por parte del usuario.

## **1.5 Conclusiones**

En este capítulo se ha indagado sobre la Geología como ciencia, que estudia la composición del planeta, su ubicación y actuación en el universo así como sus ramas principales. Se ha mostrado además cuánto conocimiento geológico se ha acumulado durante años de estudio en nuestro país, y las diferentes tecnologías que han sido y son usadas para la gestión de datos geológicos. Se ha definido el concepto de Metadato como un término que va más allá de catalogación y que ha sido desarrollado con fines mayormente informáticos, y su implementación deberá seguir estándares específicos teniendo en cuenta las características particulares de cada uno.

El perfil cubano de Metadatos Geológicos, desarrollado por la ONRM, está basado en la ISO 19115 aplicando las características del estándar a las especificidades particulares de nuestro país. De esta manera la implementación de un sistema informático para la creación y edición de metadatos, deberá ser lo más amigable posible, en cuanto a interfaz de usuarios,

navegable y comprensible, así como de fácil operatividad, mantenimiento y entendimiento por parte del usuario, según se ha observado en las características de otras soluciones que se han dado a problemas similares.

## CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR.

### Introducción

En este capítulo se hace un análisis de los diferentes entornos de trabajo en los que se desarrollan sistemas similares. Además se realiza una exposición de tendencias, técnicas, tecnologías y metodologías utilizadas en la actualidad. De esta manera, el objetivo de este capítulo es definir bien cuales serán las tecnologías, que se usarán, se escogerá una metodología para el desarrollo del software, así como las herramientas informáticas para el modelado del proyecto.

### 2.1 Metodologías de desarrollo.

Una metodología es un modo de realizar el análisis sistemático de los principios racionales que guían los procesos de adquisición de saberes epistémicos (saberes en filosofía, ciencias, etc.), así como de los procesos de configuración de los contenidos de una ciencia o disciplina en sus estructuras, articulación y conexiones temáticas, etc. Es decir, una metodología es una observación sobre los contenidos de una ciencia, teoría o sobre los métodos de ésta. **(10)**

Una metodología describe un proceso de desarrollo de software, en el que se definen *Quién* está haciendo *Qué*, *Cuándo* y *Cómo* alcanzar un objetivo determinado. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas que el desarrollo actual de las tecnologías permite.

Un proceso debe servir como guía a todos los participantes en el proyecto: clientes, usuarios, desarrolladores y directores ejecutivos; debe estar actualizado, y disponible de manera que todos los interesados puedan entenderlo; debería también evolucionar durante mucho tiempo y limitar su alcance a las realidades que permitan las tecnologías, herramientas, personas y patrones de organización. **(14)**

Entre las metodologías que actualmente existen para el desarrollo de software se encuentran: Extreme Programin (XP), que es utilizada para proyectos de corto plazo y corto equipo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

**Características de XP**, la metodología se basa en:

**Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de manera que se prevé hacia el futuro, que podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.

**Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

**Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Por otra parte, **Microsoft Solution Framework (MSF)** es flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. Se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

**MSF** tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MÉTRICA Versión 3 contempla el desarrollo de Sistemas de Información para las distintas tecnologías que actualmente están conviviendo y los aspectos de gestión que asegurarán que un Proyecto cumple sus objetivos en términos de calidad y coste. **(11)**

La metodología Rational United Process presenta varias ventajas sobre las anteriores tecnologías, por ejemplo, XP da la posibilidad al cliente de cambiar o quitar requisitos en cualquier momento, ya que el cliente forma parte del equipo, los requerimientos pueden ser

actualizados en todo momento. RUP gestiona los requerimientos de manera diferente, ya que hay un flujo de trabajo que es para la captura de requisitos, que tiene su mayor esfuerzo en la fase inicial, lo cual posibilita que no se tenga que cambiar el sistema a cada momento y que se trabaje basados en un destino fijo (hitos).

XP se basa también en que cuanto antes haya alguna versión funcionando mejor y se irá añadiendo mejoras y funcionalidades a medida que hagan falta, en RUP hay un momento preciso, en la fase de construcción, donde ya debe haber al menos una versión del sistema, esto posibilita que el proceso ya haya transitado por un correcto análisis y diseño, y la implementación de esta primera versión se realiza basada en todos los requerimientos analizados hasta el momento, lo cual es mucho mejor ya que el cliente puede ver algo semejante a lo que será el producto final.

Por otra parte podemos decir que MSF deja en segundo plano las elecciones tecnológicas. RUP si tiene en cuenta esto con un nivel razonable de importancia, las características de las tecnologías, sus ventajas y desventajas, ayudan en la definición del modelado del proyecto.

Además podemos decir que para elegir una metodología a seguir depende del tipo de proyecto y la cultura que predomina en los programadores, en nuestro caso se puede concluir, basándonos en estos dos aspectos: por ser un proyecto relativamente corto y contar con el conocimiento de RUP, la principal metodología que se estudia en nuestra universidad, es la que se selecciona para la realización de este trabajo.

A continuación se realiza una breve descripción de la metodología de desarrollo de software, RUP.

### **2.1.1 Metodología RUP (Rational Unified Process).**

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través de UML (Lenguaje Unificado de Modelado), y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vió la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo. Se utilizó para crear los diagramas teniendo en cuenta sus flujos de trabajo.

En el año 1995, Ivar Jacobson, James Rumbaugh y Graddy Booch, en aquel entonces trabajando para la compañía Rational Software, comienzan a crear primero el Lenguaje

Unificado de Modelado (UML) y luego en el Proceso Unificado Racional (RUP), Jacobson había trabajado en el año 1967 en el desarrollo de la nueva generación de conmutadores telefónicos controlados que la compañía Ericsson estaba desarrollando. Para ello inventó diagramas de secuencia y desarrolló diagramas de colaboración. También aplicó diagramas de transición de estado para describir el flujo de mensajes entre los componentes. Booch ingeniero eléctrico, obtuvo su licenciatura en las fuerzas aéreas en 1977 y posteriormente en 1979 obtiene su maestría en la universidad de california, luego del trabajo con Jacobson y Rumbaugh continuó escribiendo libros sobre el Análisis y Diseño Orientado a Objetos, proponiendo en los mismos la adición de clases para simplificar los códigos. James Rumbaugh, doctor en ciencias de la computación, continuó escribiendo libros sobre UML y RUP, en el 2003 se trasladó a la IBM y se retiró en el 2006.

### **Características del RUP:**

- Creado por Ivar Jacobson, James Rumbaugh y Grady Booch.
- Unifica los mejores elementos de metodologías anteriores.
- Preparado para desarrollar grandes y complejos proyectos.
- Orientado a Objetos.
- Utiliza el UML como lenguaje de representación visual. **(11)**

Como se ha dicho anteriormente, un proceso de desarrollo define *Quién* está haciendo *Qué*, *Cómo* y *Cuándo*. En la figura Fig. 21 se presenta los principales elementos de RUP, y como se relacionan entre sí.

RUP divide en cuatro fases el desarrollo del software:

### **Fases:**

- Conceptualización: se define la visión del proyecto.
- Elaboración: determinar la arquitectura óptima.
- Construcción: obtener la capacidad funcional y operacional inicial.
- Transición: obtener el despliegue del sistema.

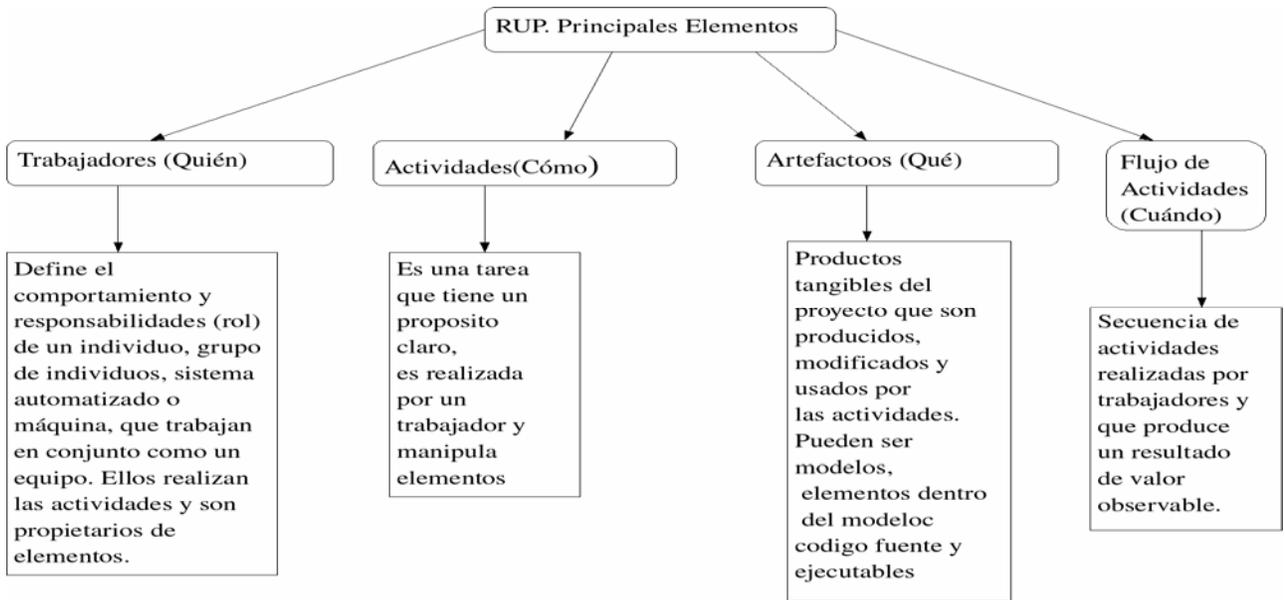


Figura 3 Principales elementos del RUP.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

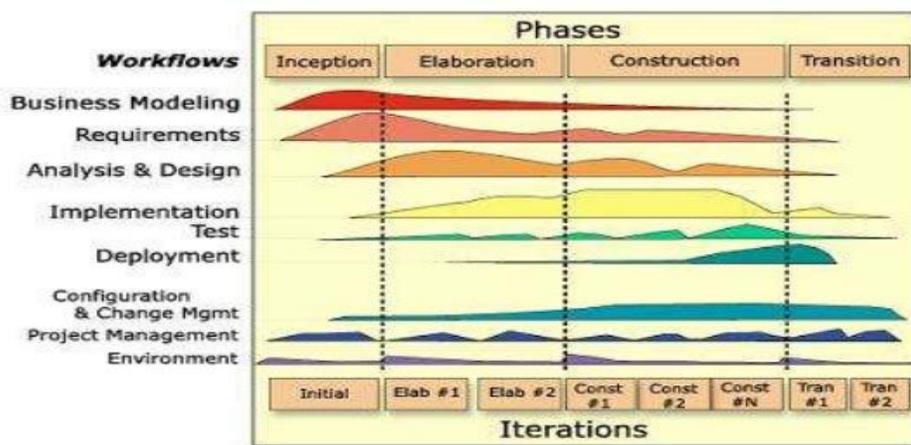


Figura 4 Flujos de trabajo de RUP; términos en inglés.

## Flujos de trabajo:

*...de ingeniería.*

- Modelamiento del negocio: para entender las necesidades del negocio.
- Requerimientos: para convertir las necesidades del negocio a un sistema automatizado.
- Análisis y diseño: para acoplar los requerimientos dentro de la arquitectura del software.
- Implementación y Prueba: para crear un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Instalación: para desplegar el producto.

*...de apoyo.*

- Administración del proyecto: para administrar horarios y recursos.
- Administración de configuración y cambios: para administrar los cambios en las versiones del proyecto.
- Ambiente: para administrar en ambiente de desarrollo.

## 2.2 Lenguaje de Modelado UML.

Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. El UML esta compuesto por diversos elementos gráficos que se combinan para conformar diagramas, debido a que es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. **(12)**

UML será de gran utilidad, puesto que RUP usa sus diagramas para las representaciones visuales.

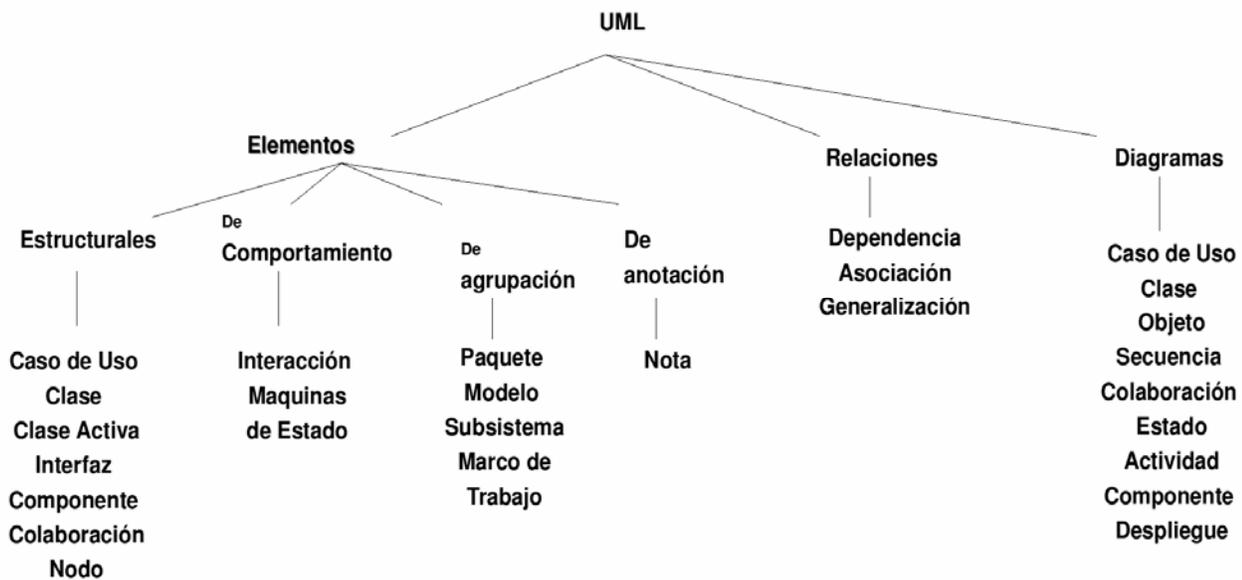


Figura 5 Vocabulario de UML.

## 2.3 Sistema Operativo

Los sistemas operativos están formados por un conjunto de programas destinados a permitir una administración eficaz de los recursos de un ordenador, gestionan todo el hardware desde los niveles más básicos, permitiendo así la comunicación con el usuario, la gestión de los archivos, y la administración de las tareas entre otras funciones.

En la actualidad las dos corrientes de desarrollo de software: Software Libre, Software Propietario, compiten por aumentar la usabilidad, eficiencia y seguridad de sus sistemas operativos. El proyecto PNICG, en el que se desarrolla este trabajo, funciona sobre las políticas del software libre, teniendo en cuenta las ventajas económicas que ofrece esta corriente, y el desarrollo tecnológico avanzado que ha llegado a obtener.

### 2.3.1 Linux.

Linux es un sistema operativo que tiene múltiples ventajas sobre los demás: la primera es que es gratis, o sea no hay que pagar ningún tipo de licencia para su uso, al igual que para la mayoría de los software que corren sobre el. Es un Sistema Operativo multitarea, permite ejecutar varios programas (procesos) simultáneamente, es además multiusuario, pueden estar conectados varios usuarios a la misma maquina. Todo el código fuente está disponible,

incluyendo el núcleo completo y los controladores (*drivers*), las herramientas de desarrollo y los programas de usuario; además todo esto se puede distribuir libremente.

Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito. Este sistema operativo posee una política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.

### **2.3.2 Ubuntu.**

Ubuntu será la distribución que se usará, pues la misma cuenta con una interfaz bastante amigable y fácil de usar. El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar contra la seguridad del sistema.

Este sistema posee además un buen gestor de paquetes para la instalación de programas, códec, etc., además de abundante documentación online. Todos los lanzamientos de Ubuntu se proporcionan sin costo alguno.

## **2.4 Lenguajes de Programación**

### **2.4.1 JAVA.**

Java es, por encima de todo, un lenguaje de programación. Pero sus características le están convirtiendo en la piedra angular de lo que puede ser la gran revolución en el mundo informático: acabar con el dominio de Microsoft y su sistema operativo Windows. El lenguaje Java es un lenguaje lo bastante potente para desarrollar aplicaciones en cualquier ámbito. Es un lenguaje de propósito general con el que se puede escribir desde un *applet*<sup>5</sup> para una página Web hasta una aplicación financiera en modo texto sin ninguna conexión a Internet. El objetivo principal de Java era conseguir un entorno de desarrollo de software que fuera

---

<sup>5</sup> Es un componente de una aplicación que se ejecuta en el contexto de otro programa.

independiente de la plataforma de ejecución. Esto se logra a través de de la Maquina Virtual, puede ser usada cualquier versión desarrollada por la compañía SUN Microsystems, (se usará en el presente trabajo la versión 1.6).

Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el *garbage collector* (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un hilo (*thread*) de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- Aritmética de punteros.
- Sintaxis de las referencias.
- Registros (*struct*)
- Definición de tipos (*typedef*)
- Macros (*#define*)
- Necesidad de liberar memoria (*free*)

Aunque, en realidad, lo que hace es eliminar las palabras reservadas (*struct*, *typedef*), ya que las clases son algo parecido.

Estas características de portabilidad, versatilidad, además de la importante reducción de los errores más comunes de programación lo convierten en un candidato prácticamente invencible a la hora de desarrollar cualquier tipo de aplicación.

## 2.4.2 XML

Por sus siglas en inglés *Extensible Markup Language*, es un metalenguaje extensible por etiquetas, desarrollado por el *World Wide Web Consortium (W3C)*. Es una simplificación y adaptación del lenguaje SGML y permite desarrollar la gramática de lenguajes específicos.

Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

XML es extensible, lo que significa que una vez diseñado un lenguaje y puesto en explotación, igual es posible extenderlo, con la adición de nuevas etiquetas de manera que los usuarios de la versión anterior puedan entender el nuevo formato. Es además, un contenedor de datos ideal, utilizando los esquemas.

Con XML se puede representar eficientemente casi cualquier dato de manera que pueda ser leído por otros usuarios.

XML es internacional, proporciona flexibilidad, sencillez de utilizar, y se pueden generar formatos estándares de documentos XML de forma simple.

En conclusión, la aplicación que se desarrollará en este proyecto, atiende a los requerimientos de los clientes y entre los cuales está la creación de una aplicación de escritorio. Para ello se utilizará el lenguaje de programación java, como decisión del proyecto PICG, y el lenguaje de marcado XML, como contenedor de datos, siguiendo el estándar para la gestión de metadatos, ISO-19115, en que se especifica este requerimiento y además ofrece un diccionario de los elementos que existen en los ficheros XML de Metadatos. Por su parte la ISO 19139, propone un esquema XML, para los ficheros de metadatos geológicos, este estándar será utilizado a manera de consulta, siguiendo más a fondo la propuesta del perfil de metadatos cubano.

## **2.5 Entorno de Desarrollo Integrado. (IDE).**

### **2.5.1 NetBeans.**

NetBeans es un IDE libre, es decir un Entorno de Desarrollo Integrado de código abierto para desarrolladores de software. Creado por una organización con el mismo nombre ahora propiedad de la compañía Sun Microsystem, empresa creadora de tecnologías como Java y Solaris. NetBeans permite obtener todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales y también para la Web, usando para ello C/C++ y Java, se ejecuta en varias plataformas, como Windows, Linux, Mac OS X y Solaris, solo se requiere la instalación de la máquina virtual de java, resulta muy fácil su instalación y uso.

La versión 6.0 de Netbeans, posee muchas mejoras con respecto a las anteriores, una reestructuración completa de la infraestructura del editor, soporte para lenguajes adicionales, y un proceso muy simple de instalación-configuración que permite al usuario ajustarlo a sus necesidades.

En NetBeans todas las aplicaciones Java son interpretadas, es decir que primero tiene que leerse el programa y después convertirlo en ejecutable.

NetBeans es extensible, es decir que terceros pueden desarrollar módulos para ejecuciones de nuevas tecnologías, para uso personal o grupal. Los módulos o extensiones se obtienen a través de un asistente que se conecta a un servidor universal que contiene módulos generales para el público, y la opción de conectarse a terceros y obtener el módulo en extensión \*.nbm e instalarlo en el NetBeans. La creación de módulos en Netbeans es muy sencilla ya que existen muchas librerías de APIs<sup>6</sup> para desarrollo.

---

<sup>6</sup> Un API representa una.

## **2.5.2 Eclipse.**

Eclipse es un IDE similar a NetBeans desarrollado por Eclipse Foundation, es multiplataforma, es decir que se puede encontrar instaladores, para Windows, Mac o Linux. Eclipse solo puede instalarse en diferentes plataformas si se posee el instalador de cada una. Eclipse es muy rápido debido a que corre en el lenguaje nativo de cada plataforma.

Al igual que NetBeans, Eclipse es también extensible. Al haber más desarrolladores trabajando en él existen más módulos (extensiones). Los módulos o extensiones se encuentran en servidores independientes, lo que hace un poco engorroso su búsqueda, pues cuando se necesita uno en específico se debe tener primero la URL.

Se ha tomado la decisión de desarrollar el software en Netbeans 6.0 a pesar de la velocidad, y la gran variedad de módulos que ofrece Eclipse, puesto que el NetBeans es menos pesado en memoria y trae más librerías incluidas y posee gran facilidad para el desarrollo de aplicaciones de escritorio por su amplia biblioteca de componentes visuales.

## **2.6 Herramienta de Diseño CASE.**

### **2.6.1 Rational Rose**

Es la herramienta líder en el mundo de modelación visual para el proceso de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes.

Rational Rose es la herramienta más poderosa del mercado, se integra además con aplicaciones desarrolladas con tecnología *J2EE* y *.NET*.

IBM Rational Rose Enterprise es uno de los productos más completos de la familia Rational Rose. Todos los productos de Rational Rose dan soporte a UML, pero no son compatibles con las mismas tecnologías de implementación. Rational Rose Enterprise es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic y ofrece un lenguaje de modelado común que agiliza la creación del software.

### **Incluye también estas funciones:**

- Soporte a modelos de análisis, ANSI C++, Rose J y Visual C++ según el documento "*Design Patterns: Elements of Reusable Object-Oriented Software*".
- Los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos más granular.
- Soporte para compilación y descompilación de las construcciones más habituales de Java 1.5.
- Generación de código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código.
- Soporte para Enterprise Java Beans 2.0
- Funciones de análisis de calidad de código.
- Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web.
- Modelado en UML para diseñar bases de datos, que integra los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos.
- Creación de definiciones de tipo de documento DTD en XML.
- Integración con otras herramientas de desarrollo de IBM Rational.

### **2.6.2 Visual Paradigm**

Esta herramienta CASE (*Computer-Aided Software Engineering*), ha sorprendido gratamente por el trabajo que se puede llegar a desarrollar básicamente.

Esta herramienta soporta hasta la fecha UML 2.1 completo, permite realizar ingeniería tanto directa como inversa pues a partir de un modelo relacional en SQL Server, MySQL y otros servidores de Bases de Datos es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Para gestionar la persistencia y el mapeo de estas clases con la base de datos utiliza Hibernate para Java y NHibernate en el caso de un proyecto .Net.

Esta herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos

como HTML o .Pdf, y permite el control de versiones. Cabe destacar igualmente su robustez, usabilidad y portabilidad.

Durante su instalación se puede escoger entre sus paquetes una gran variedad de servicios adicionales como un pequeño Ambiente de Desarrollo, llamado *Smart Development Environment (SDA)*, que utiliza las características principales de UML, y las integra con varios de los IDE favoritos como NetBeans, Eclipse, JBuilder, entre otros, acelerando así el proceso de generación de código. Otro de los servicios es el de Arquitectura Visual para Base de Datos, llamado *DB Visual Architect (DBVA)*, es un ambiente de desarrollo que hereda las características visuales de UML y crea un canal de comunicación, desde el punto de vista de ingeniería, entre las bases de datos relacionales y los diagramas de clases, de esta manera genera objetos persistentes java, a partir de las bases de datos relacional. Esto reduce el esfuerzo en el desarrollo del software significativamente.

En el presente trabajo se diseñará el software con Visual Paradigm porque ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML, posee versiones además para Linux, y además, permite generar esquemas de XML (*XML Schema*), que será muy útil a la hora de trabajar con los archivos XML (requerimiento de la aplicación).

## **2.7 Patrones de Diseño**

*"Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma"*

*Christopher Alexander,  
Universidad de Oxford, 1977.*

Los Patrones de Diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Estos se dividen en tres grandes categorías:

- Patrones Creacionales: Solucionan problemas de creación de instancias. Ayudan a encapsular y abstraer dicha creación.
- Patrones Estructurales: Solucionan problemas de composición (agregación) de clases y objetos.
- Patrones de Comportamiento: Soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan. **(12)**

#### **Un patrón de diseño es:**

1. Una solución estándar para un problema común de programación.
2. Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
3. Un proyecto o estructura de implementación que logra una finalidad determinada.
4. Una manera más práctica de describir ciertos aspectos de la organización de un programa.
5. Conexiones entre componentes de programas.
6. La forma de un diagrama de objeto o de un modelo de objeto.

#### **Ventajas:**

1. Contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. La importancia de la reutilización del diseño no es despreciable ya que ésta nos provee de numerosas ventajas: reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia de nuestros diseños y nos proporciona un considerable ahorro en la inversión.
2. Mejoran la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la calidad percibida por el usuario.
3. Incrementan nuestro vocabulario de diseño, ayudándonos a diseñar desde un mayor nivel de abstracción. **(12)**

Entre los patrones de diseño se encuentra el **patrón Experto**: su función principal es asignar una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad.

Este patrón distribuye las tareas y responsabilidades a las clases que poseen la información necesaria para ejecutarlas, esto ayuda a mantener la encapsulación de la información y el bajo acoplamiento lo que conlleva a sistemas robustos y de mantenimiento mucho más fácil.

El **patrón Creador**, por su parte, guía la asignación de responsabilidades relacionadas a la creación de objetos y su intento básico es encontrar un creador que necesite estar conectado al objeto creado en un evento particular. Este patrón también beneficia el bajo acoplamiento, el mantenimiento y la reusabilidad del código.

En el caso del **patrón Controlador** asigna la responsabilidad de manejar un sistema de mensajes a una clase representando todo el sistema o todos los negocios de la organización. Una misma clase controlador debe ser usada para todos los eventos del sistema de un caso de uso, que llevará en todo momento la razón del estado de la información del proceso, y tal información es útil para identificar eventos fuera de secuencia.

Un defecto común en el diseño de controladores es darles muchas responsabilidades. Normalmente, un controlador delegaría a otro objeto el trabajo que necesita ser realizado mientras coordina la actividad.

Delegando la responsabilidad de operación del sistema a un controlador, se incrementan los elementos, de la lógica del proceso de negocio, que puedan ser reutilizados para el futuro en otras aplicaciones.

## **2.8 Patrones de Arquitectura**

La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de:

- Definir los módulos principales
- Definir las responsabilidades que tendrá cada uno de estos módulos
- Definir la interacción que existirá entre dichos módulos:
- Control y flujo de datos
- Secuenciación de la información

- Protocolos de interacción y comunicación
- Ubicación en el hardware

### **2.8.1 Modelo Vista Controlador (MVC)**

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. **(12)**

El Modelo es todo acceso a datos, y las funciones que llevan lo que llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos, este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo modelo. **(12)**

La Vista, en una aplicación Web, es el HTML y lo necesario para convertir datos en HTML. O sea muestra la información del modelo al usuario además tienen un registro de su controlador asociado. Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML (Wireless Markup Language), sin tener que tocar más que una parte completamente delimitada del código. **(13)**

El Controlador es lo que une la Vista y el Modelo, por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las entradas del usuario, recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.), contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. De este modo, el código que "hace algo" está perfectamente separado del código dedicado a crear HTML.

### **2.8.2 Por Capas**

Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden

hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar qué puede reutilizarse, y proporciona una estructura que nos ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Principales estilos de arquitecturas estratificadas de las aplicaciones distribuidas contemporáneas:

- Arquitecturas de dos niveles.
- Arquitecturas de tres niveles.
- Arquitecturas de n niveles.

### **Arquitectura de tres niveles.**

Para enfrentarse a estos temas, la comunidad de software desarrolló la noción de una arquitectura de tres niveles. La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido.

Estas tres capas son:

#### **La capa de la Presentación.**

Esta capa reúne todos los aspectos del software que tienen que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general.

#### **La capa de Negocio (Capa Lógica).**

Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican.

#### **La capa de Datos.**

Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes, por lo que también se le denomina la capa de las Bases de Datos.

Teniendo en cuenta que el producto final del presente trabajo será una aplicación de escritorio, se eligió el patrón capas. Pues el patrón modelo-vista-controlador (MVC) suele ser muy eficiente en aplicaciones Web por las características antes mencionadas, sin embargo el patrón capas permite una mejor abstracción y encapsulamiento de los datos, lo que lo hace óptimo para desarrollar aplicaciones con características similares.

## **2.9 Otros Programas Informáticos Utilizados.**

Como procesador de texto se utilizó el Microsoft Word 2007 como Procesador de Texto.

## **2.10 Conclusiones**

En este capítulo se evidencia la necesidad del desarrollo de una aplicación sobre el sistema operativo Linux y específicamente la distribución Ubuntu, por ser software libre por las características económicas y legales que brinda. Además debe ser una aplicación de escritorio, desarrollada en el lenguaje de programación Java implementado sobre el IDE Netbeans, con un soporte de datos proporcionado por XML, siguiendo los estándares de metadatos. La Metodología de desarrollo es RUP, la cual es Orientado a Objetos y para el Modelamiento visual se usará UML, el cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Se seguirán además los patrones de diseño Experto, Creador y Controlador para asegurar la mayor eficiencia en la construcción del software. La arquitectura será por capas, asegurando un bajo acoplamiento, una alta cohesión entre las futuras capas del software y una mayor facilidad de mantenimiento.

## CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

### **Introducción**

En el presente capítulo se comenzará a modelar la propuesta para dar solución al problema científico tratado. Siguiendo la metodología RUP que propone un esfuerzo inicial en el modelado del Negocio y la captura de los Requerimientos. En estos flujos de trabajo son desarrollados un conjunto de artefactos como el modelo de casos de uso del negocio, donde se identifican actores y casos de uso del negocio, que se describen minuciosamente, realizándose los diagramas de actividades. Se desarrolla además la captura de los requerimientos funcionales y no funcionales. Se construye el modelo de sistema donde se identifican los actores del futuro sistema y los casos de uso del sistema que guiarán el desarrollo del software.

### **3.1 Modelo de Negocio**

La ONRM, y todas sus oficinas afines en todo el país, administran un gran número de estudios e investigaciones geológicas cuyos datos se almacenan en bases de datos, tanto en formato digital como duro, en archivos y registros. De esta manera se hace necesaria la creación de metadatos que permitan una eficiencia mayor en la gestión de la información. El perfil de metadatos desarrollado por el Programa Geodato ya está siendo probado en varias de las oficinas de forma manual, donde la captura de los metadatos es realizada por especialistas, quienes manejan los formularios y variables del metadato. Los datos son facilitados por geólogos responsables de las bases de datos que serán documentadas. Estos metadatos conformarán el catálogo nacional de metadatos geológicos en la ONRM.

En el estudio de la documentación de los procesos de captura de metadatos se obtuvieron las siguientes reglas del negocio:

1. El usuario de la aplicación entrará, como datos mínimos necesarios para la creación del Metadato, los que se recogen en el Núcleo del Metadato. Que esta conformado por los campos o variables obligatorias en los modelos obligatorios. (En el documento “Procedimiento

para la documentación del Metadato”, los modelos obligatorios se encuentran marcados como “(obligatorio)”.

2. Una vez que el usuario decida entrar datos en un modelo “no obligatorio” deberá llenar al menos las variables o campos obligatorios de ese modelo. (Las variables o campos obligatorios se encuentran marcados como “\*”).

El modelo del negocio tiene como objetivo identificar los principales procesos y describirlos en secuencias lógicas de actividades, para alcanzar una visión general de todo el proceso que se desea automatizar. Se identifican también los actores como los beneficiados por cada proceso y los trabajadores quienes gestionan estos beneficios.

### 3.1.1 Actores y trabajadores del negocio

Actor	Justificación
Geólogo	Responsable de la base de datos que se quiere documentar, proveedor de los datos necesarios para la creación del metadato geológico.
Trabajador	
Especialista	Especialista en la creación de metadatos, responsable de la correcta documentación de los mismos, a partir de los datos facilitados por el Geólogo.

Tabla 5 Definición de actores y trabajadores.

### 3.1.2 Procesos de negocio

En la visión general del problema se identifica un único proceso: La creación de metadatos geológicos.

El Geólogo responsable de base de datos que se documentará, solicita la creación de su metadato, provee de los datos necesarios. El especialista ofrece los formularios

correspondientes, recibe los datos, y asegura el correcto formato realizando así la documentación del metadato.

### 3.1.3 Diagrama de Casos de Uso del Negocio

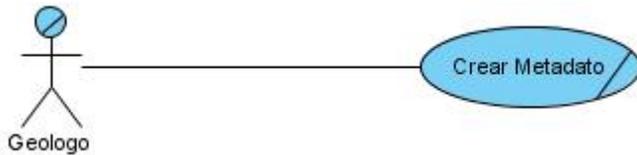


Figura 6 Diagrama de Casos de Uso del Negocio.

### 3.1.4 Descripción textual de los Casos de Uso de Negocio

Caso de Uso:	Crear Metadato	
Actores:	Geólogo	
Trabajadores:	Especialista	
Resumen:	El geólogo entrega los datos de su Rasgo Geológico, el especialista se asegura que los datos tengan el formato correcto y sigan el estándar ISO19115. Crea el Metadato.	
Precondiciones:		
Flujo Normal de Eventos		
"Crear metadato"		
Acción del Actor	Respuesta del Negocio	
1. Solicita Crear un metadato	1.1 Ofrece los formularios para crear el metadato	
2. Entrega los datos correspondientes	2.1 Verifica las reglas de entrada de datos.	
	3.1 Crea el Metadato.	

Flujos Alternos	
Acción del Actor	Respuesta del Negocio.
Pos condiciones	El Metadato creado debe contener al menos el formulario del Núcleo del Metadato, con todos sus campos llenos.

Tabla 6 Descripción textual de los Casos de Uso de Negocio.

### 3.2 Descripción del Sistema Propuesto

Durante la captura de requerimientos en entrevistas con el cliente, y mediante el estudio de los estándares para captura y documentación de los metadatos geológicos se obtuvo la siguiente lista:

#### Requerimientos Funcionales:

1. El sistema debe permitir la gestión de los Metadatos.
  - 1.1 Debe permitir la creación de los Metadatos.
  - 1.2 Debe permitir la actualización de los Metadatos
2. El sistema debe actualizar los Nomencladores.

#### Requerimientos No Funcionales:

##### *Requerimientos de Software:*

1. Se debe usar Sistemas Operativos Windows a partir de su versión Windows 2000 o Linux.
2. Se debe usar la máquina virtual de java versión 6.0.

##### *Requerimientos de diseño e implementación:*

3. Se seguirá el estándar ISO 19115 para la documentación del Metadato.
4. Se utilizará JAVA como lenguaje de programación y XML.
5. Se utilizará como entorno de desarrollo Net Beans 6.0
6. Se utilizará como herramienta CASE el Visual Paradigm.
7. Se usará la biblioteca JDOM 1.1 de ORG para el tratamiento de los ficheros XML.

##### *Requerimientos de apariencia o interfaz:*

8. La apariencia visual deberá ser muy similar a los formularios del documento “Procedimiento para la Documentación del Metadato”.

*Requisitos de hardware:*

9. Se necesita una memoria física mínima de 64 Mb de RAM, y al menos 98 Mb libre en el disco duro.

### 3.2.1 Descripción de los actores

Actor	Justificación
Especialista	Especialista en la creación de metadatos, responsable de la correcta documentación de los mismos, a partir de los datos facilitados por el Geólogo. Puede crear, y actualizar Metadatos, y actualizar los nomencladores.

Tabla 7 Descripción de los actores

### 3.2.2 Casos de Uso del Sistema

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema. Puede decirse que una especificación funcional contesta a la pregunta: ¿Qué debe hacer el sistema? La estrategia de los casos de uso puede describirse, añadiendo tres palabras al final de esta pregunta: ¿... para cada usuario? Estas tres palabras albergan un significado importante, nos fuerzan a pensar en términos de importancia para el usuario y no solo en términos de funciones que sería bueno tener. Sin embargo, los casos de uso no son solo una herramienta para especificar los requisitos de un sistema, también guían su diseño, implementación y pruebas; guiando así todo el proceso de desarrollo. (14)

Después de realizada la captura de los requerimientos, estos se agrupan en dos casos de uso a modelar:

#### 1. Gestionar Metadato:

De forma general este es el caso de uso principal del sistema sobre el cual gira toda la aplicación, ya que es el objetivo primario de la misma la creación o generación y

actualización de metadatos debidamente revisados, libre de errores, y en formato XML siguiendo los estándares ya establecidos de antemano por la norma ISO 19115.

## 2. Actualizar Nomencladores.

Los nomencladores o listas de códigos son valores predefinidos de algunos campos de los modelos que forman el metadato, los cuales son emitidos por la ONRM de forma centralizada. Este caso de uso actualizará los nomencladores de la aplicación.

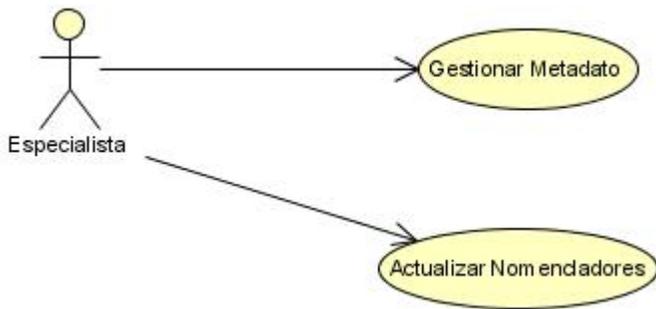


Figura 7 Diagrama de Casos de Uso del Sistema.

### Descripción textual del caso de uso: Gestionar Metadato

Caso de Uso:	Gestionar Metadato
Actores:	Especialista
Resumen	El especialista decide crear o actualizar un metadato. Si decide " crear " el sistema recibe los datos que entra el actor, asegurando el correcto formato de los mismos. Si decide Actualizar, el sistema carga el fichero deseado y el especialista actualiza los datos, finalmente los datos son guardados en un fichero XML.
Precondiciones	
Referencia	
Prioridad	Crítico
Flujo Normal de Eventos	
"Gestionar metadato"	
Acción del Actor	Respuesta del Sistema

1	Solicita gestionar un Metadato	1.1 Ofrece opciones “Crear” o “Actualizar”
2	a) Escoge opción Crear. <i>Sección 1.</i>  b) Escoge opción Actualizar. <i>Sección 2.</i>	
<b>Sección 1: Crear Metadato</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1.	Entra los datos en los formularios	2.1 El sistema ofrece los formularios del metadato.  3.1 Comprueba el correcto formato de los datos. 3.2 Crea el Metadato.
<b>Sección 2: Actualizar Metadato</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
3	Escoge Metadato para actualizar.	2.1 El sistema ofrece los metadatos disponibles para cargar.
4	Actualiza los datos.	3.1 Carga los datos en los formularios.  4.1 Guarda los cambios.
<b>Prototipo de Interfaz</b>		
Poscondiciones	El Metadato creado o actualizado debe cumplir con las reglas del negocio: todos los formularios y campos obligatorios llenados correctamente.	

Tabla 8 Descripción textual del caso de uso: Gestionar Metadato.

### Descripción textual del caso de uso: Actualizar Nomencladores

Caso de Uso:	Actualizar Nomencladores	
Actores:	Especialista	
Resumen	El especialista recupera el fichero XML de nomencladores nuevo, y los carga en el sistema.	
Precondiciones	Debe existir al menos un fichero XML de nomencladores de metadatos.	
Referencia		
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Actualizar Nomencladores"		
Acción del Actor	Respuesta del Sistema	
1. Solicita actualizar los nomencladores.	1.1 Ofrece los ficheros de nomencladores disponibles.	
2. Escoge el fichero que desea cargar.	2.1 Actualiza los nomencladores en el sistema	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
Prototipo de Interfaz		
Poscondiciones	Los nomencladores deben quedar actualizados.	

Tabla 9 Descripción textual del caso de uso: Actualizar Nomencladores

### **3.3 Conclusiones**

Después de un profundo análisis acerca del funcionamiento de la captura de metadatos geológicos en la ONRM, se ha realizado un estudio de la visión general del negocio, se definió un proceso único en el cual se gestionan los metadatos geológicos.

En el proceso de la captura de requerimientos, realizado mediante entrevistas con el cliente y la revisión de documentos afines con la creación de metadatos geológicos, así como su estandarización según las normas internacionales, se definieron varios requisitos funcionales, entre los cuales se destacan la captura de metadatos y su almacenamiento en ficheros XML. Entre los requisitos no funcionales son de mayor interés: que el sistema pueda ser ejecutado en diferentes sistemas operativos como Windows y Linux, el sistema debe seguir los estándares para metadatos geológicos y además como otros requisitos, se definieron las plataformas de diseño e implementación.

## CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

### Introducción

En este capítulo se realiza la construcción del sistema, partiendo de una fase de elaboración previa en la que se realiza el análisis y el diseño guiado por los casos de uso identificados en la captura de requerimientos en la fase de inicio.

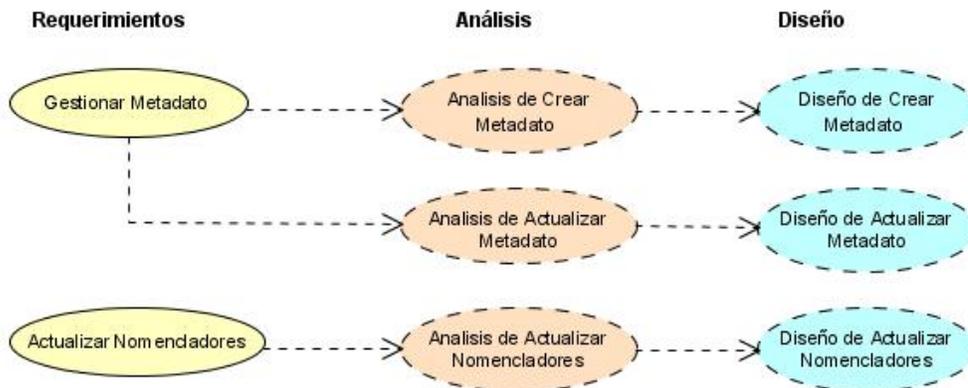


Figura 8 Proceso guiado por los casos de uso.

En el Análisis se desarrollan diagramas de clases y de interacciones en los que se puede comprender la comunicación entre clases e interfaces que serán el punto de partida para el diseño.

Además se realiza el diseño de la estructura Metadato, adaptando las características del perfil desarrollado por Cuba de la norma ISO 19115. Este diseño contiene los ocho paquetes UML que propone el perfil cubano, y ha sido modelado orientado a objetos teniendo en cuenta cada una de las estructuras de datos contenidas en los modelos.

Se realiza además la construcción del software donde se obtiene por primera vez código fuentes y ejecutables que serán las primeras versiones del futuro producto. Se muestran además los diagramas de componentes, de esta manera se obtiene una visión de la arquitectura general del sistema.

## 4.1 Análisis

En el análisis se desarrolla una serie de artefactos que forman parte de la primera iteración del análisis y diseño siguiendo la metodología RUP. Partiendo de la fase anterior donde se definieron dos casos de uso principales del sistema, se agrupan en dos escenarios de análisis y diseño respectivamente para el caso de uso “Gestionar Metadato”, y un escenario para el caso de uso “Gestionar Nomencladores.”

### 4.1.1 Escenario: Gestionar Metadato.- Crear Metadato

Se identifican una clase interfaz para la comunicación con el usuario, dos clases controladoras que desarrollan en negocio –validación de los datos y reglas del negocio- y el acceso a los datos respectivamente, una clase entidad llamada Metadato, formada por los ocho paquetes de clases que serán descritos más adelante.



Figura 9 Diagrama de Clases: Capturar Metadato.

La comunicación durante el tiempo de vida del caso de uso Crear Metadato, se realiza mediante seis mensajes o estados para obtener el objetivo final. El diagrama de secuencia se puede ver en el Anexo 1.

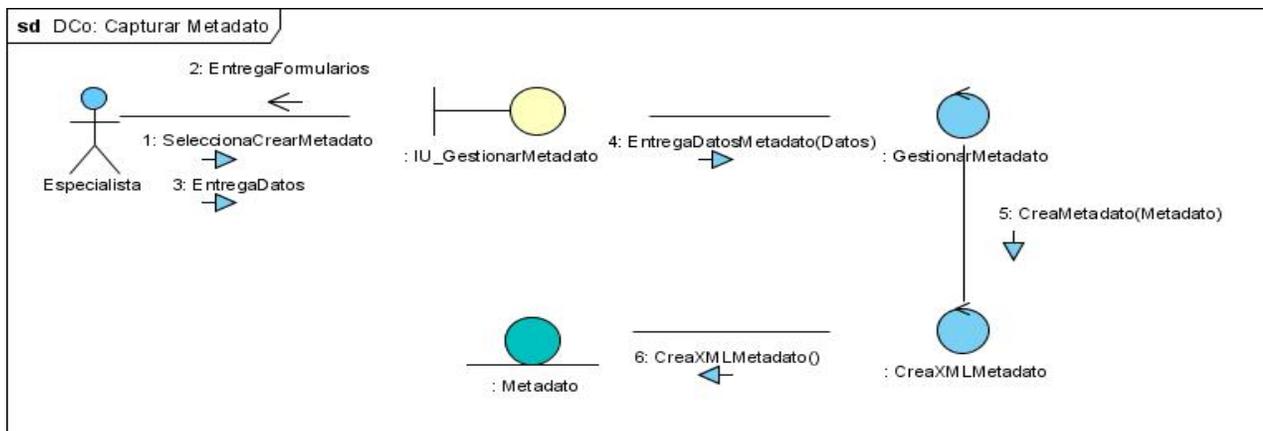


Figura 10 Diagrama de colaboración: Crear Metadato

#### 4.1.2 Escenario: Gestionar Metadato.- Actualizar Metadato.

Para el caso de uso Actualizar Metadato, siguiendo las características del patrón de diseño por capas, se han identificado también cinco clases: interfaz de usuario para comunicación con el cliente, tres clases controladoras para llevar el negocio y para el acceso a los datos y la clase entidad Metadato. El diagrama de secuencia se puede ver en el Anexo 1.

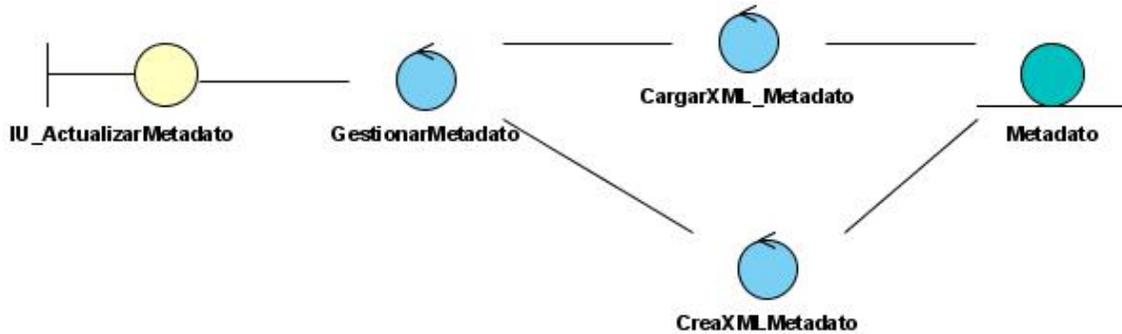


Figura 11 Diagrama de clases: Actualizar Metadato.

La colaboración entre las clases se realiza de manera general mediante trece mensajes o estados que aparecen de forma secuencial logrando así el objetivo final.

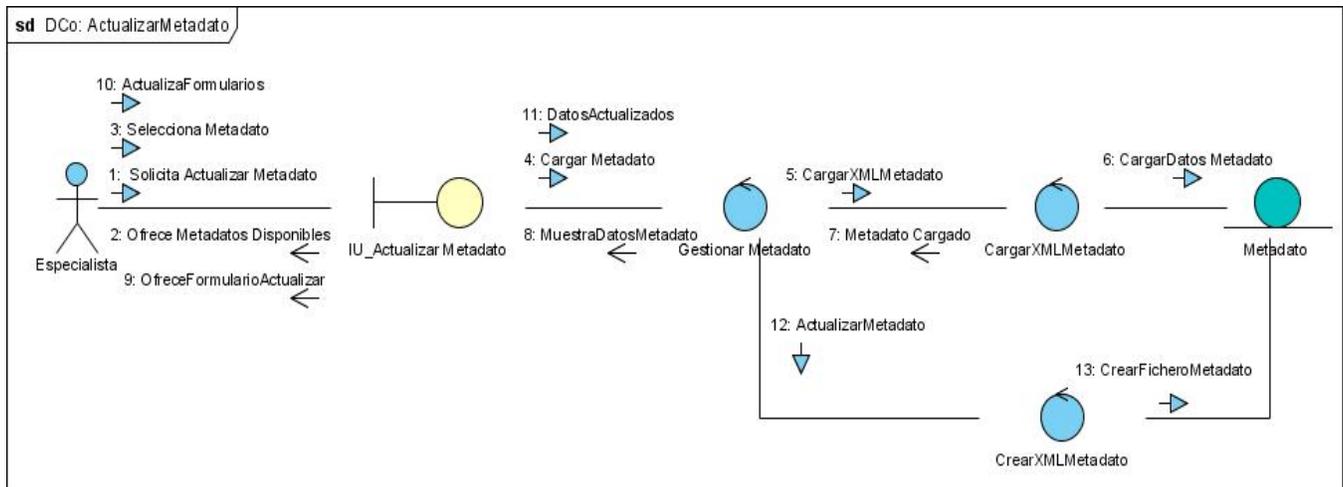


Figura 12 Diagrama de Colaboración: Actualizar Metadato.

### 4.1.3 Escenario: Actualizar Nomencladores

En el caso de uso Actualizar Nomencladores y teniendo en cuenta que los nomencladores se gestionan en formato XML, se han identificado tres clases: una interfaz de usuario, una clase controladora, encargada de actualizar el fichero nomencladores en la aplicación, y la clase entidad del nomenclador. El diagrama de secuencia se puede ver en el Anexo 1.



Figura 13 Diagrama de clases: Actualizar Nomencladores.

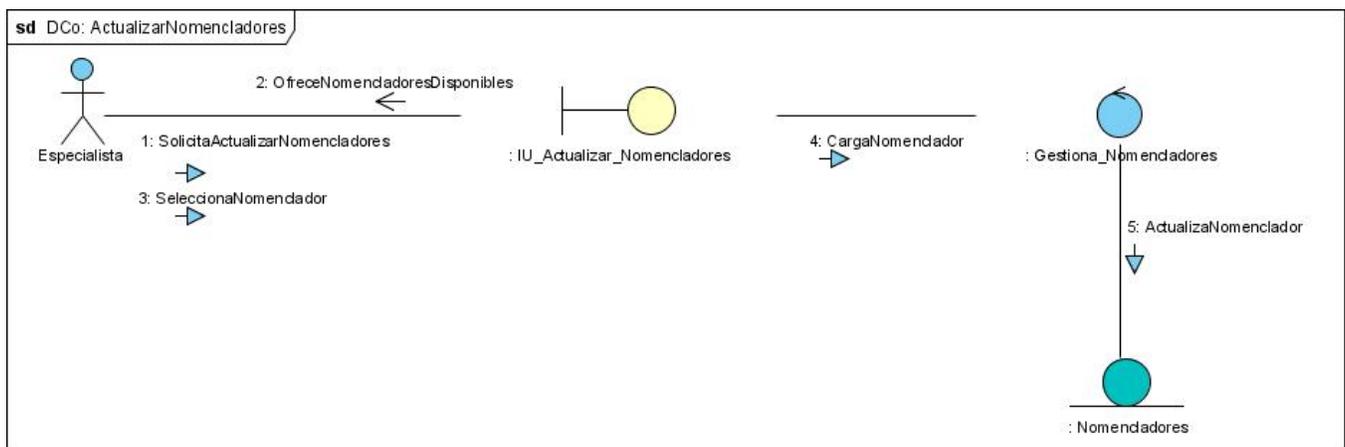


Figura 14 Diagrama de colaboración: Actualizar Nomencladores

## 4.2 Diseño

En el diseño los artefactos creados serán todos los escenarios de diseño (por casos de uso), y un paquete de clases identificadas como entidades que modelaran cada uno de los paquetes que componen el Metadato.

### 4.2.1 Diagrama de clases: Metadato

Como se ha dicho anteriormente el metadato está formado por ocho paquetes UML seleccionados por el perfil cubano, y compuestos por modelos según el contenido. A continuación se relacionan los ocho paquetes que han sido modelados orientados a objetos

según las estructuras de datos que aparecen en los modelos del documento: Procedimiento para la documentación del Metadato del Programa Geodato.

### Metadato.

Como se ha visto los ocho paquetes están anidados a una clase superior, en la cual está contenida en el metadato. En el siguiente diagrama se puede tener una visión de cómo están formadas y relacionadas todas las clases del metadato.

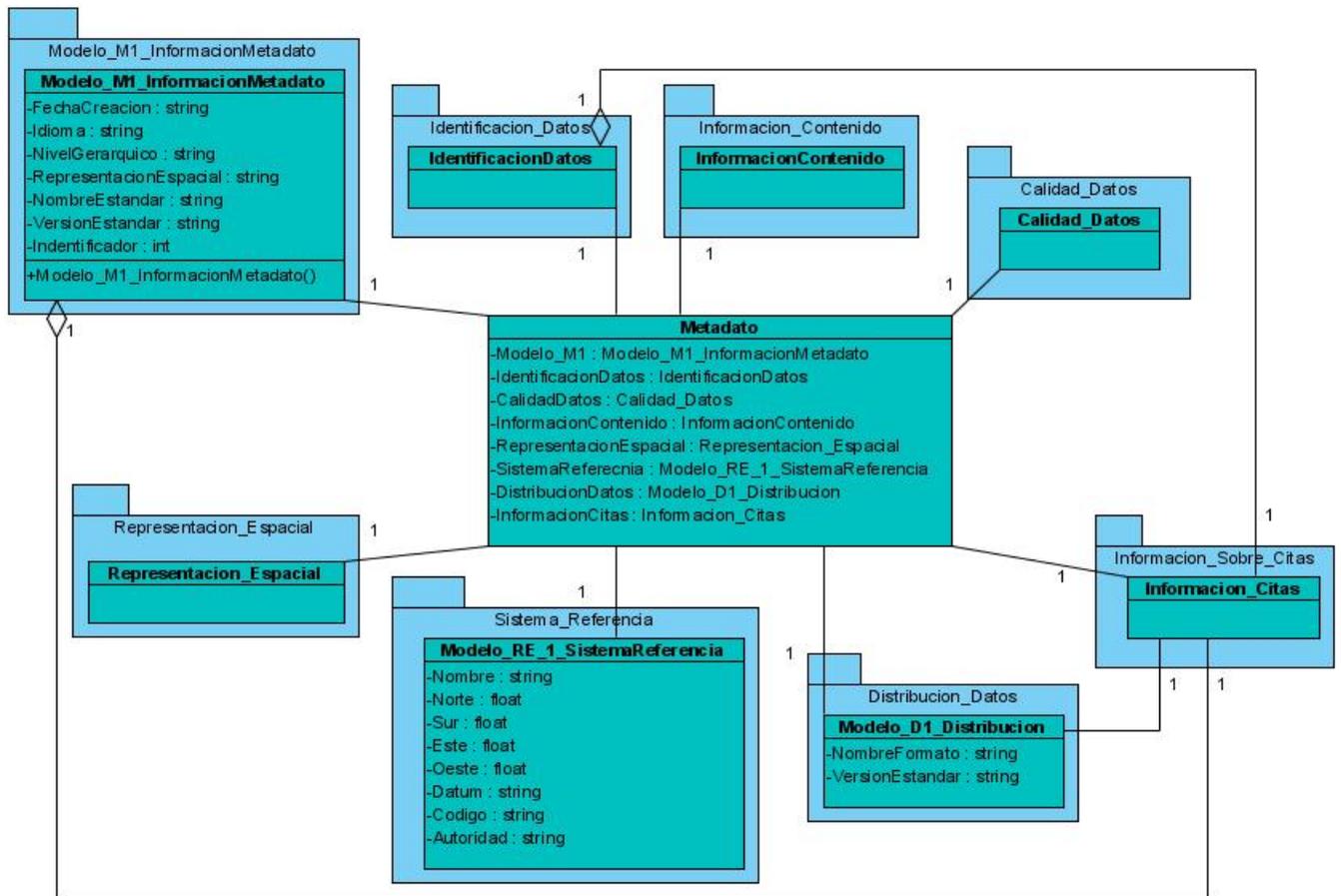


Figura 15 Diagrama de clases entidades del paquete: Metadato.

### Paquete: Información del Metadato.

El paquete Información del Metadato cuenta con un sólo modelo llamado Modelo M-1 Información del Metadato que recogen la información general del metadato, y en el están implícitos un conjunto de coordenadas para ubicar geográficamente los datos de la base de datos que se documenta, estas coordenadas contienen a su vez un conjunto de vértices.

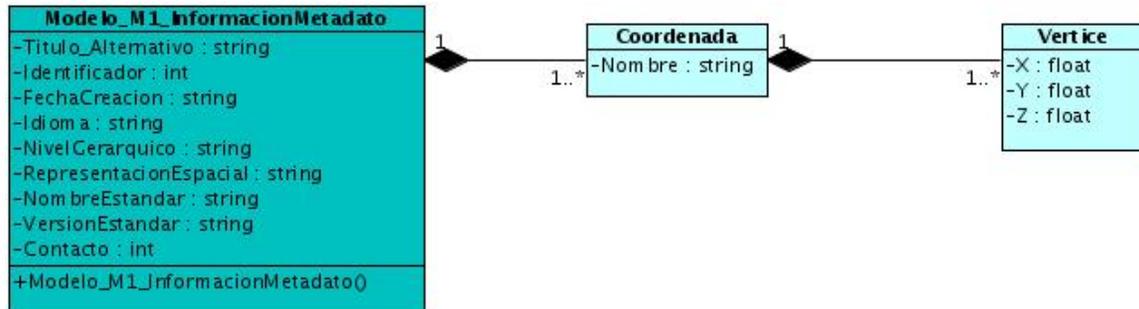


Figura 16 Diagrama de clases del paquete: Información del Metadato.

## Paquete: Identificación de los Datos

El paquete Identificación de los Datos contiene cuatro modelos:

1. Modelo I-1 Identificación: contiene además de sus variables que describen los datos de la base de datos que se documenta, un conjunto de límites geográficos.
2. Modelo I-2 Palabras Claves: contiene un conjunto de palabras claves definidas de esa manera para la base de datos que se documenta.
3. Modelo I-3 Mantenimiento: contiene datos sobre el mantenimiento de los datos de la base de datos que se documenta.
4. Modelo I-4 Restricciones: contiene las restricciones legales y de seguridad para los datos de la base de datos que se documenta.

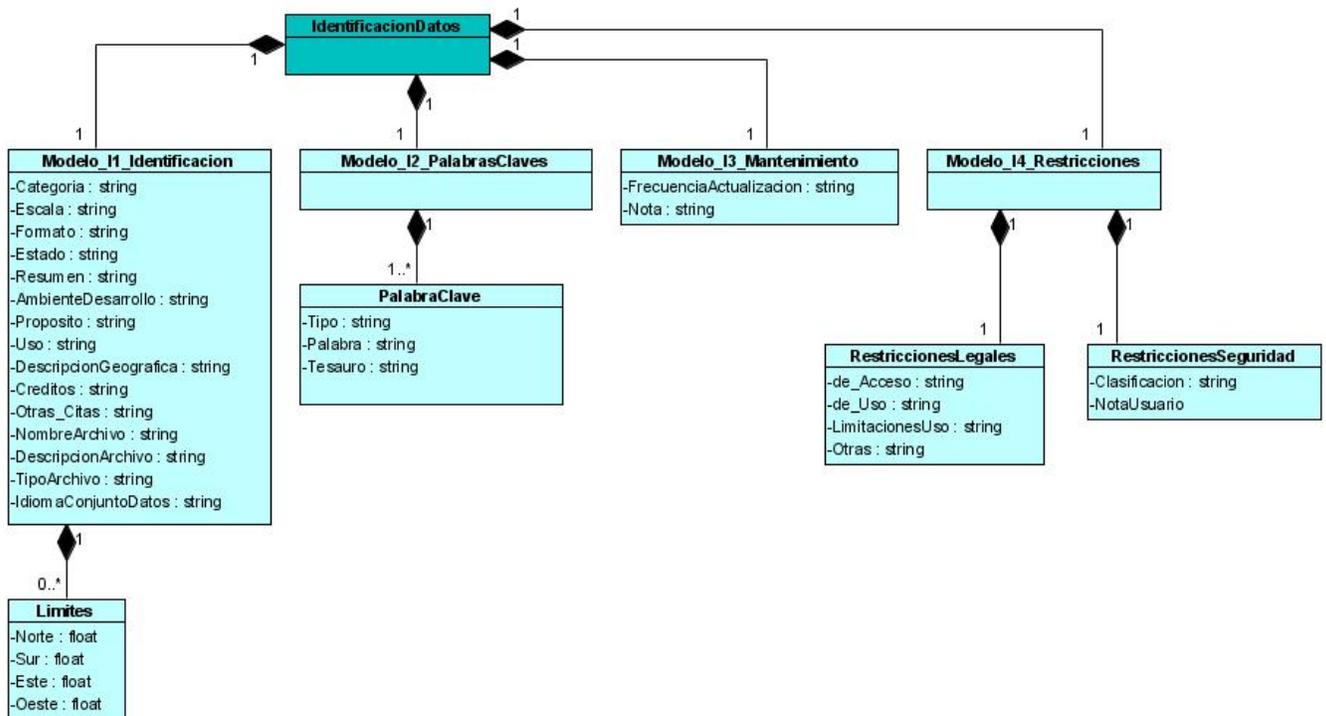


Figura 17 Diagrama de clases del paquete: Identificación de los Datos.



## Paquete: Información del Contenido

El paquete cuenta con tres modelos que describen el contenido de la base de datos geológica que se documenta.

1. Modelo CO-1 Definición de Objetos: contiene la definición de todos los objetos de la BD que se documenta, según la tecnología BDARV, que contienen a su vez, un conjunto de variables virtuales y elementales presentes en cada objeto.

2. Modelo CO-2 Estructura de la Base de Datos: contiene un conjunto de los nombres de las tablas existentes y cada uno de sus atributos.

3. Modelo CO-3 Relaciones en la Base de Datos: contiene un conjunto de las relaciones entre las tablas existentes en la base de datos y los índices de las tablas relacionadas.

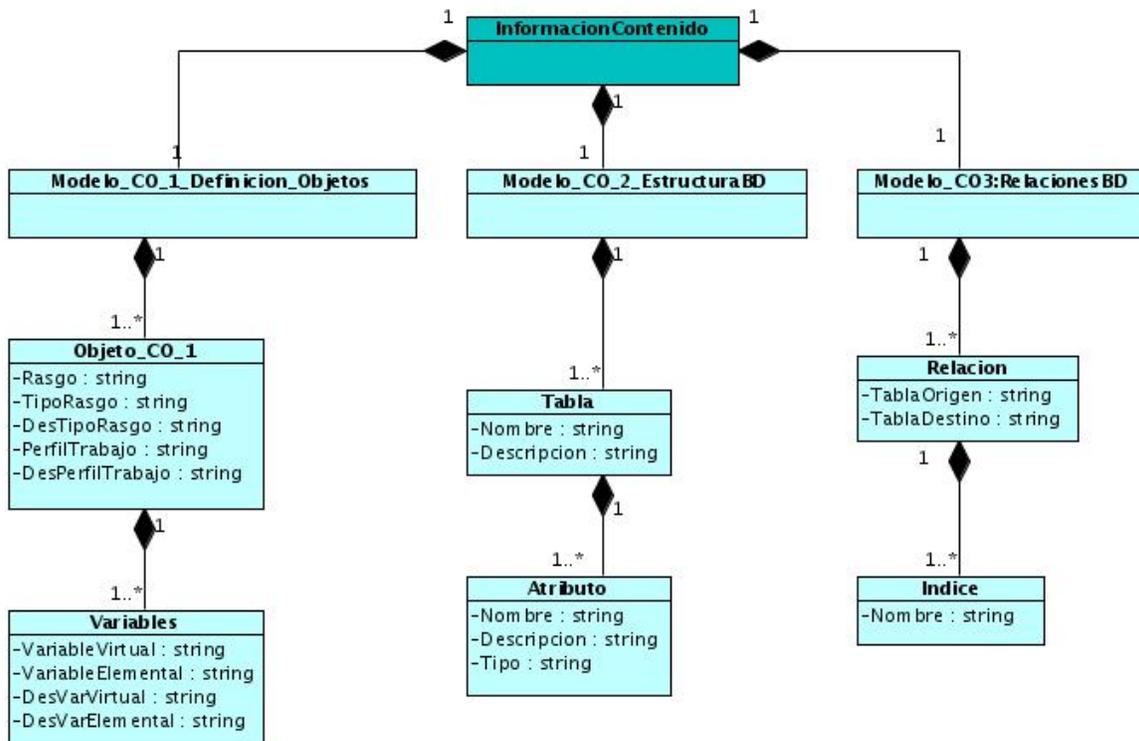


Figura 18 Diagrama de clases del paquete: Información del Contenido.



**Paquete: Calidad de los Datos.**

El paquete contiene cinco modelos que recogen los datos sobre la calidad y linaje del conjunto de datos que se documenta:

1. Modelo CA-1 Reporte de Completamiento.
2. Modelo CA-2 Reporte de Consistencia Lógica.
3. Modelo CA-3 Reporte de Precisión de la Posición.
4. Modelo CA-4 Reporte de Precisión Temática.

Estos cuatro modelos representan los cuatro elementos de calidad según el nombre de cada modelo, cada uno de ellos contiene un conjunto de subelementos con sus respectivas características.

5. Modelo CA-5 Linaje: Este modelo recoge el linaje de los datos mediante una serie de pasos del proceso de creación del conjunto de datos en la base de datos que se quiere documentar.

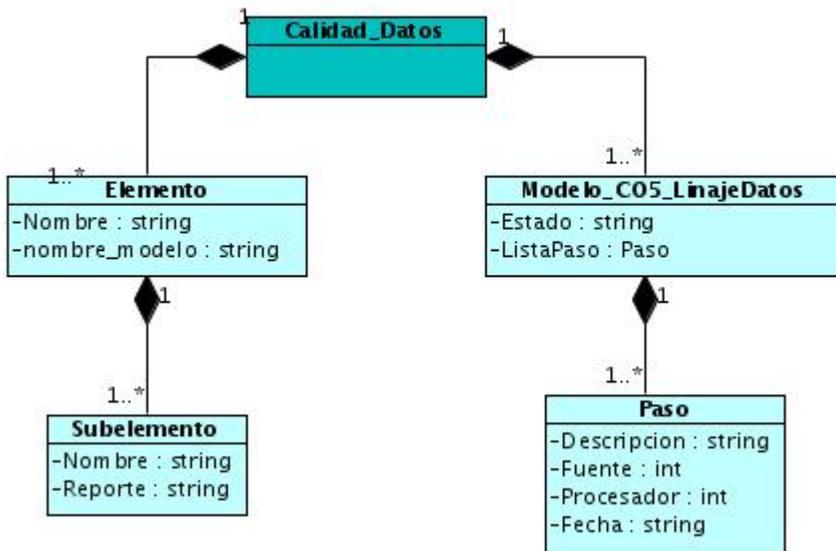


Figura 19 Diagrama de clases del paquete: Calidad de los Datos.

## Paquete: Representación Espacial

Este paquete contiene los datos de representación espacial Ráster y Vectorial en dos modelos:

1. Modelo RE-1 Ráster
2. Modelo RE-2 Vectorial

Este último contiene un conjunto de objetos topológicos con algunas características.

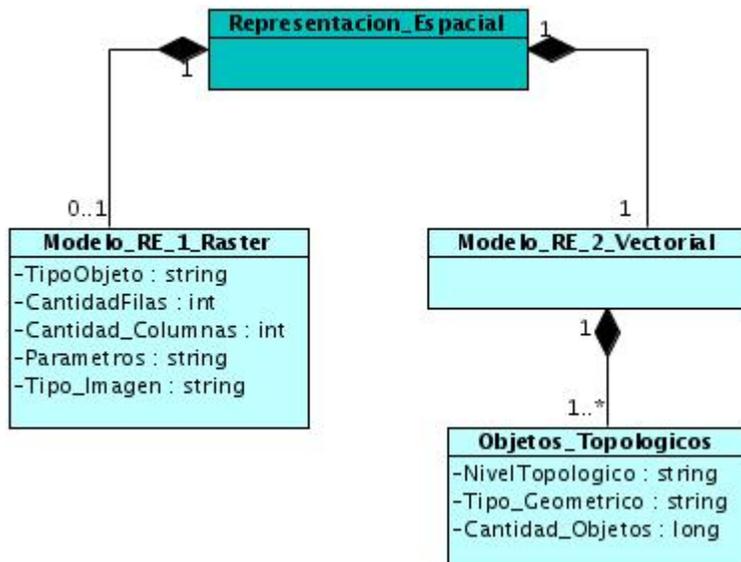


Figura 20 Diagrama de clases del paquete: Representación Espacial.



### Paquete: Sistema de Referencia

El paquete contiene un solo modelo llamado Modelo RE-1 Sistema de Referencia en el cual se describe el sistema de referencia con el que están documentados los datos y sus límites geográficos. Además contiene los datos sobre la Proyección y el Elipsoide donde se encuentran referenciados los datos de la base de datos que se documenta.

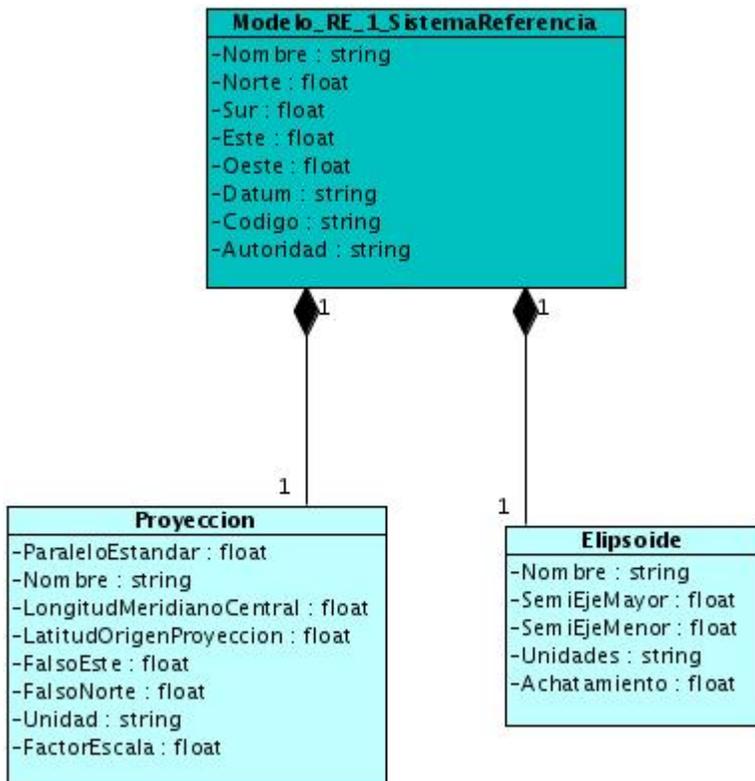


Figura 21 Diagrama de clases de paquete: Sistema de Referencia.

**Paquete: Distribución de los Datos.**

Este paquete contiene un solo modelo llamado Modelo D-1 Distribución que contiene los datos sobre el proceso de pedido para obtener el conjunto de datos, y las vías de transferencia, esta última de dos formas diferentes: En Línea y Fuera de Línea.

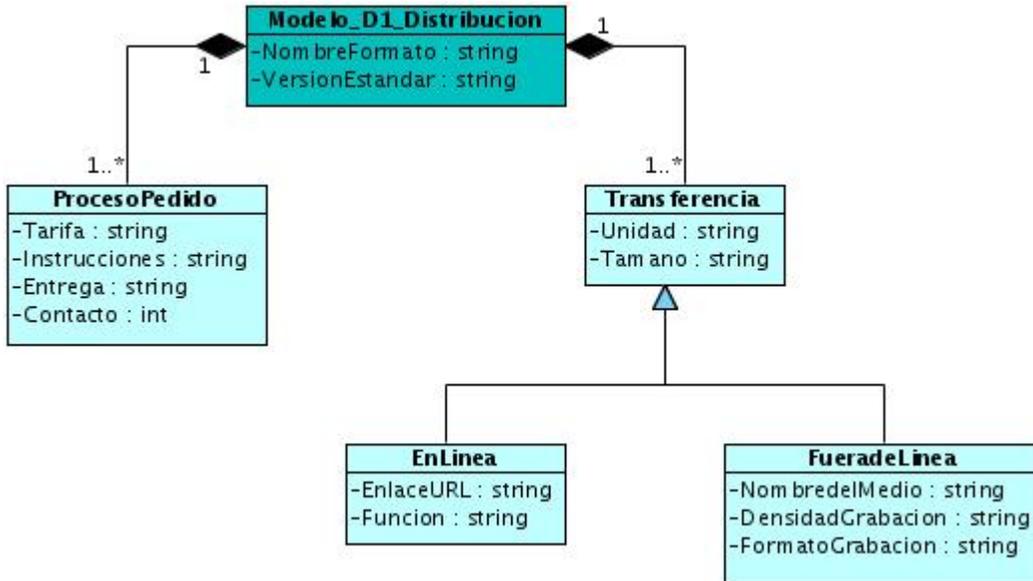


Figura 22 Diagrama de clases del paquete: Distribución de los Datos.

## Paquete: Información sobre Citas.

Este paquete contiene dos modelos que recogen los datos de las citas referenciadas durante la documentación del metadato del conjunto de datos, y sus responsables.

1. Modelo CI-1 Citas.

2. Modelo CI-2 Grupo Responsable.

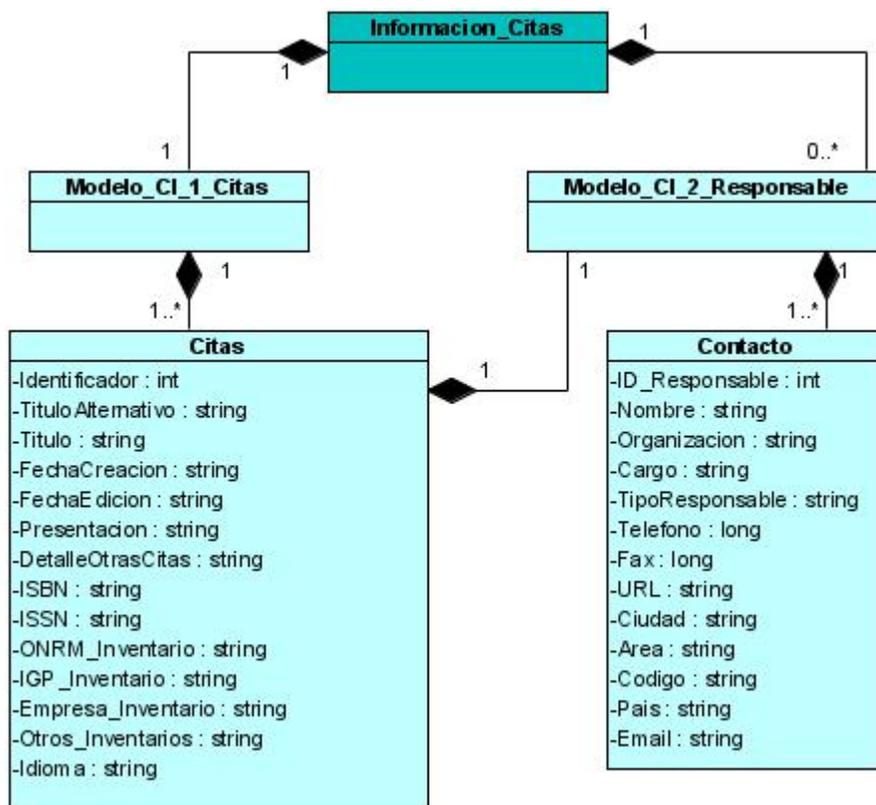


Figura 23 Diagrama de clases del paquete: Información sobre Citas.



## Diagrama de clases: Escenario Crear Metadato.

También se han diseñado las realizaciones de cada caso de uso, modelando las clases controladores y de entidad que intervienen en cada uno. Los diagrama de clase de interfaz y sus prototipos se pueden ver en los Anexos 2 y 3 respectivamente.

Como se había modelado en el análisis el escenario cuenta con dos clases controladoras. La clase *GestionarMetadato* se encarga de la construcción paso a paso de los objetos del Metadato, y de crear el documento XML, aunque para esto utiliza la clase *CrearXML\_Metadato*, que es responsable de construir uno a uno los objetos del metadatos en elementos XML, que formarán parte del documento final.

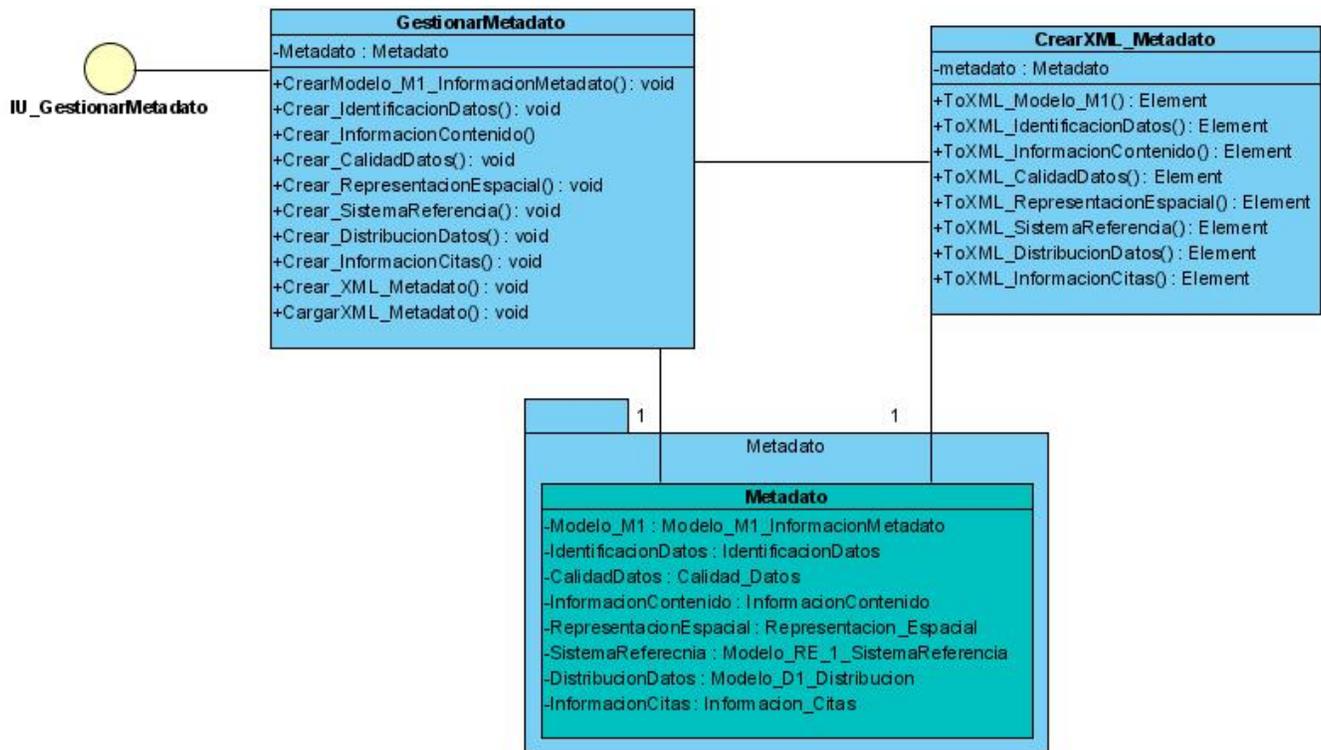


Figura 24 Diagrama de clase CU Gestionar Metadato, escenario: Crear Metadato.

#### 4.2.2 Diagrama de clases: Escenario Actualizar Metadato.

Siguiendo los modelos que se obtuvieron en el análisis se puede observar en el siguiente diagrama que existen tres clases controladoras encargadas de cargar los datos del Metadato que son leídos a través de la clase *CargarXMLMetadato* convertidos nuevamente en objetos para ser actualizados y posteriormente guardados con la clase *CrearXML\_Metadato*.

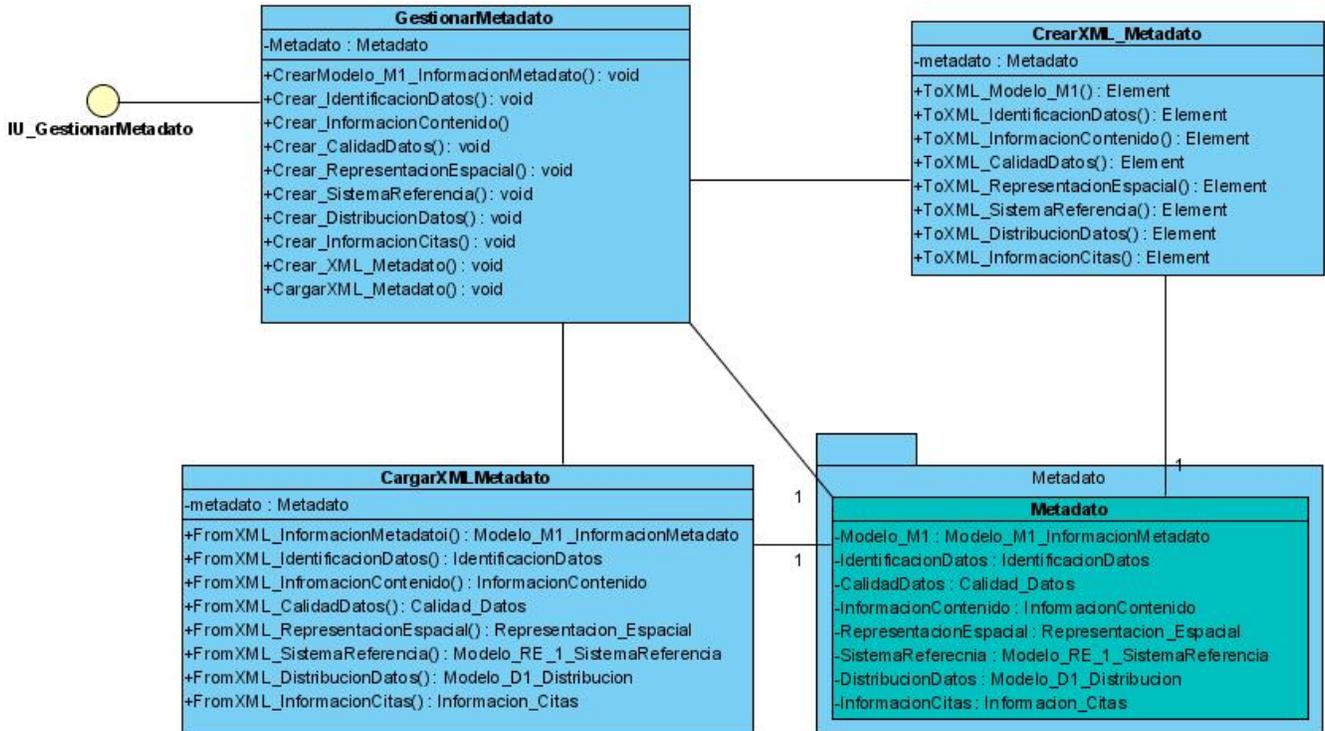


Figura 25 Diagrama de clases CU Gestionar Metadato, escenario: Actualizar Metadato.

### 4.2.3 Diagrama de clases: Actualizar Nomencladores.

En el análisis de este escenario se habían indentificado tres clases que intervienen en este caso de uso, una clase interfaz *IU\_ActualizarNomencladores*, la clase controladora *ActualizarNomencladores*, y la clase que representa la entidad *Nomenclador*. La clase controladora carga los datos del fichero XML de los nomencladores, recoge todas las listas códigos y las actualiza en la interfaz de aplicación.

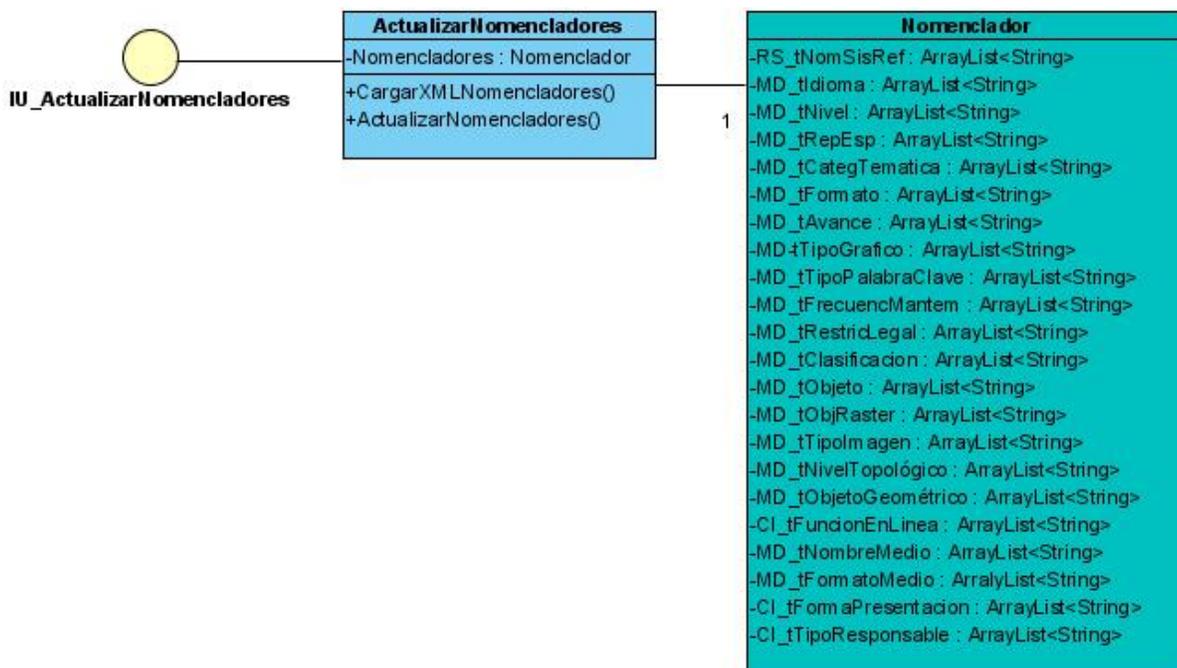


Figura 26 Diagrama de clases CU: Actualizar Nomencladores.



## **Principios de diseño**

Los siguientes principios son fundamentales para el diseño e implementación de interfaces gráficas efectivas, bien se trate de interfaces de escritorio o de la web.

Las interfaces efectivas son visualmente comprensibles y permiten errores por parte del usuario, dándole una sensación de control. Los usuarios ven rápidamente el alcance de las opciones y comprenden como alcanzar sus metas y realizar su trabajo.

Las interfaces eficientes ocultan al usuario el funcionamiento interno del sistema. El trabajo se guarda continuamente y con la opción de deshacer en todo momento cualquier paso que se haya dado. Las aplicaciones y servicios efectivos realizan el máximo trabajo requiriendo la mínima información del usuario.

### **4.3.1 Estándares de la interfaz de la aplicación**

Teniendo en cuenta que el software que se desarrolla es una aplicación de escritorio, y que su función principal es la creación y edición de ficheros metadatos, de una manera más amigable posible, en cuanto a interfaz de usuario, navegable y comprensible, de fácil operatividad y entendimiento, el diseño de aplicación seguirá algunas pautas que asegurarán los objetivos que se persiguen.

#### **Anticipación**

La aplicación deberá intentar anticiparse a las necesidades y deseos del usuario sin esperar a que el usuario busque o recuerde información o herramientas. Mostrar al usuario toda la información y herramientas necesarias para cada etapa en su trabajo ayudará en la rapidez y eficiencia en la tarea que se realiza.

#### **Autonomía**

El ordenador, la interfaz y el entorno de trabajo pertenecen al usuario que deberá tener su espacio y libertad dentro de la aplicación, en un entorno ni muy restrictivo, ni demasiado grande, un entorno explorable pero no peligroso. Los usuarios aprenden rápido y ganan confianza cuando se sienten que tienen el control del sistema.

No existe autonomía en ausencia de control; y el control no se puede tener sin información suficiente. Comunicar el estado es fundamental para que el usuario responda apropiadamente con la información disponible.

Los usuarios no tienen que buscar la información de estado. De un vistazo deberían ser capaces de hacerse una idea aproximada del estado del sistema. La información de estado

puede ser bastante sutil: Un árbol de contenido en la parte izquierda de la aplicación que mantenga informado al usuario sobre en que lugar se encuentra dentro del proceso de creación de los Metadatos y el estado en que se encuentran cada uno de los campos: *vacío, incorrecto, correcto*.

### **Daltonismo**

Si se utilizara el color para transmitir información se deberá también utilizar otros elementos complementarios para el personal con daltonismo, ya que aproximadamente un 10% de los hombres adultos sufren de esta enfermedad.

Las pistas secundarias pueden consistir en distintos tonos de gris, gráficos complementarios o etiquetas de texto

### **Consistencia**

Los siguientes principios, vistos en su conjunto, dan mucho margen para la evolución de un producto sin perjudicar los aspectos más importantes para el usuario.

Se debe mantener una consistencia estricta. En la siguiente lista aparecen los elementos de la interfaz ordenados por su necesidad de consistencia, de mayor a menor.

1. Interpretación del comportamiento del usuario. Ejemplo: los atajos de teclado deben funcionar siempre igual.
2. Estructuras invisibles.
3. Estructuras visibles pequeñas.
4. El aspecto general de una aplicación o servicio (presentación, elementos de diseño).
5. Consistencia interna.
6. Consistencia con la plataforma.

Las estructuras invisibles se refieren a objetos funcionales y de control que pueda la tener la aplicación y que no son visibles para el usuario, funcionalidades específicas que se descubren solo con el uso regular del software.

Existen otros objetos en la interfaz que se consideran pequeñas estructuras visibles, pero muchas veces no parecen controles: es posible que el usuario nunca descubra que se pueden interactuar con ellos. Su significado debería ser muy claro. Por ejemplo, se podrá hacer clic en algún pequeño icono en la esquina de la ventana para cambiar su tamaño.

### **Valores por defecto**

Los valores por defecto deberían poder ser descartados con facilidad y rapidez. Los campos de texto con valores por defecto deben aparecer seleccionados, para que el usuario sólo tenga que teclear y no seleccionar todo, borrar y escribir.

Los valores por defecto deben tener sentido. No se deben usar palabras como “por defecto”, o “campo vacío”, sino más bien debería insinuar al usuario lo que está a punto de teclear: “dd/mm/aa” para el caso de un campo de Fecha.

### **Interfaces explorables**

Imitar la seguridad, suavidad y consistencia del medio natural deberá ser una característica de la interfaz. No se debe encerrar al usuario en un único camino, mas se deberá ofrecer la ruta de menos resistencia. Esto facilita a los nuevos usuarios o a aquellos que sólo quieren hacer acabar la tarea, hacerlo rápidamente y sin esfuerzo; pero también hay que dejar posibilidades abiertas a aquellos que desean explorar sobre todo su trabajo, revisar y modificar.

Los elementos visuales estables no sólo ayudan a navegar más rápido, si no que también actúan como una referencia necesaria para sentirse seguro.

Los usuarios deberán sentir la libertad de equivocarse, son errores que a veces cometen por curiosidad, o por accidente. Sus acciones deberán ser reversibles.

#### **4.3.2 Concepción general de la ayuda**

Teniendo en cuenta que los futuros usuarios de la aplicación que se desarrolla serán los geólogos especialistas responsables de sus bases de datos geológicas, y que su intención será la de documentarlas correctamente en el catálogo nacional geológico, el sistema deberá ofrecer alguna ayuda que lo mantenga informado sobre los significados de cada uno de los campos y conceptos del metadato, así como sobre el uso del sistema y su navegación.

La solución deberá constar de una biblioteca de ayuda que brinde toda la información agrupada en los ocho paquetes de fracciones de metadatos sobre cada uno de los campos: dominio, ocurrencia y posibles valores. Deberá brindar la opción de búsqueda de contenido por palabras, así como una pequeña introducción a principiantes que los ayude a iniciarse en el uso del software.

Durante el estudio de otras soluciones se pudieron apreciar los detalles de la Ayuda, y en uno de los casos se usaba un pequeño campo de texto en la esquina inferior izquierda, en el que se leía en todo momento el significado del campo sobre el cual el usuario estaba trabajando, se

considera esta una de las funcionalidades muy útiles para la solución de ayuda de nuestro sistema.

#### 4.4 Modelo de Implementación

Para la implementación del software, se obtuvieron los diagramas de clases del diseño y los principales requisitos funcionales. Estos diagramas de clases fueron convertidos en código fuente usando la herramienta CASE Visual Paradigm, y como resultado se implementaron todas las clases de entidades persistentes del metadato y las clases controladores y de acceso a datos con todos los métodos declarados.

Durante la implementación se desarrollaron los Constructores de todos los objetos de las clases persistentes de dos maneras: por parámetros y por defecto.

Se usó además la API<sup>7</sup> JDOM (Java Document Object Model), de la ORG para el trabajo con los ficheros XML, y la biblioteca SWING de componentes visuales para Java.

A continuación se muestra el diagrama de componentes como resultado de la primera versión de implementación del sistema. Los paquetes de componentes Metadato e Interfaz Usuario se pueden ver en los Anexos

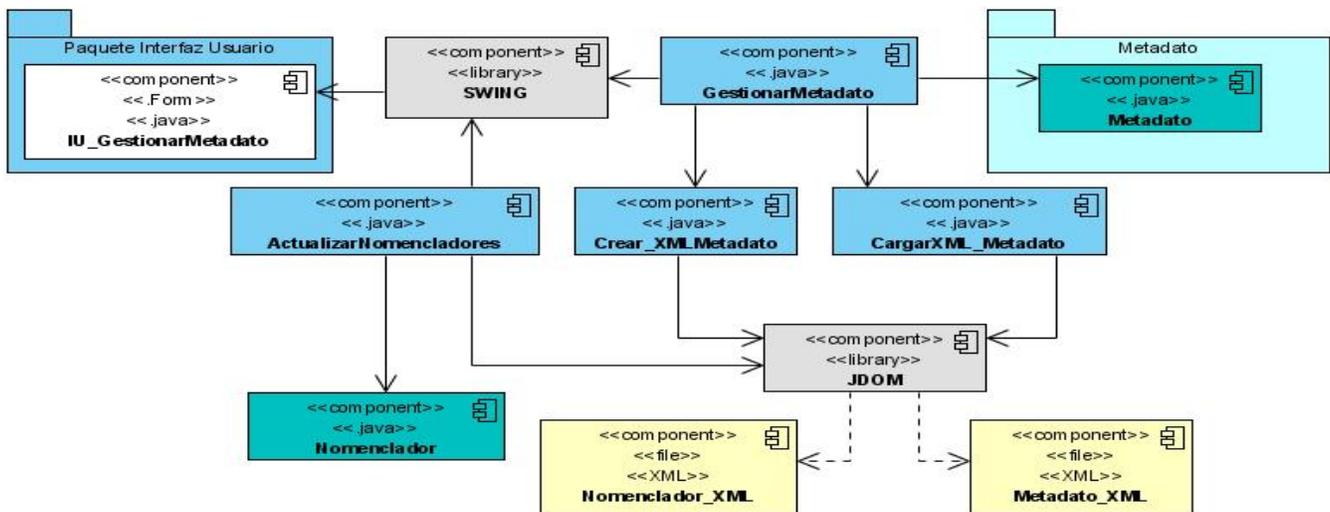


Figura 27 de componentes: Gestión de Metadatos.

<sup>7</sup> Application Programming Interface

Como se puede observar en el diagrama anterior la clase controladora *GestionarMetadatos.java* es la que realiza todas las operaciones de la Gestión de Metadatos y está relacionada con la clase de entidad *Metadato.java* que está almacenada en un paquete llamado de la misma manera y que contiene además los ocho paquetes con todas sus clases implementadas que son construidas y están relacionadas todas en *Metadatos.java*.

A continuación se muestran los ocho paquetes de implementación de estas clases, que son análogas al diseño.

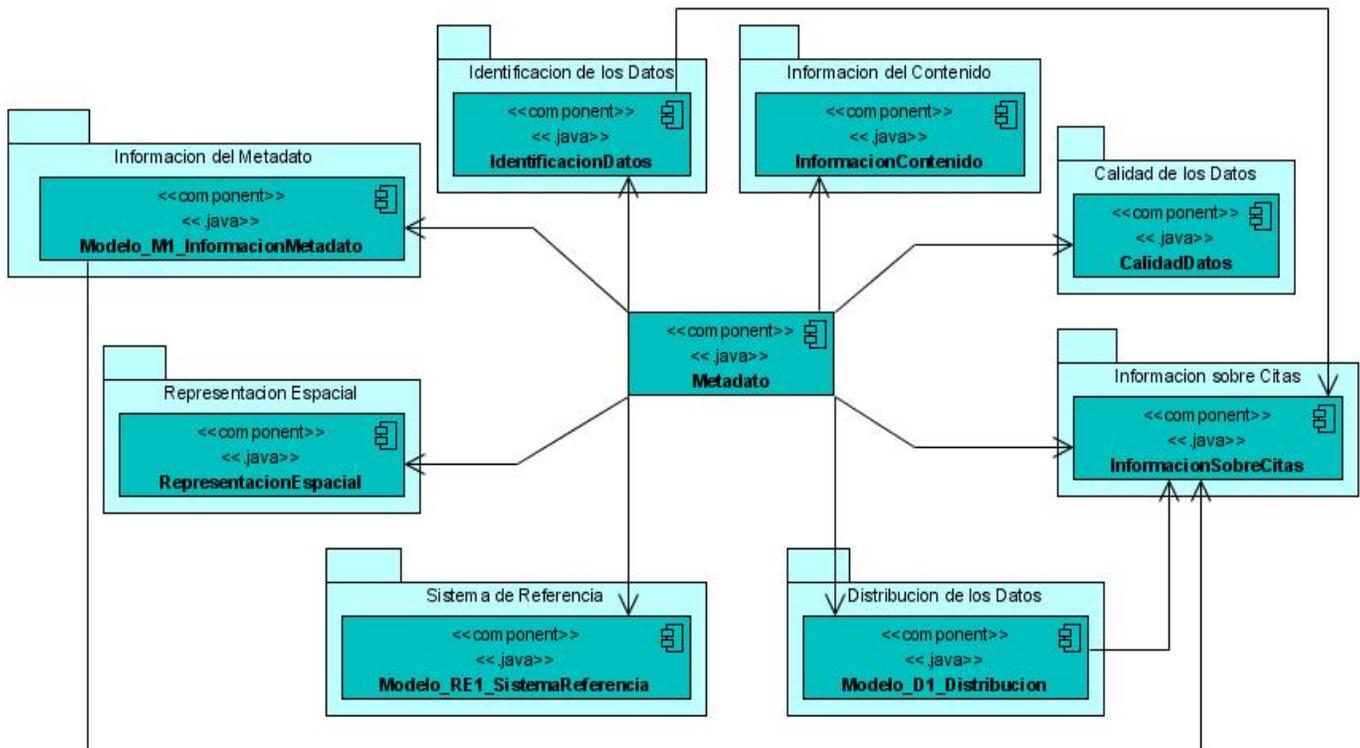


Figura 28 Diagrama de componentes: Metadato.

Se relacionan además cada uno de los paquetes de implementación con todos sus componentes y sus relaciones en el Anexo 1.

## **4.5 Conclusiones**

En este capítulo se ha realizado la construcción de la solución que se propuso, se partió de los artefactos que se confeccionaron en la fase de inicio como el diagrama de casos de uso del sistema y los diagramas de actividades y desde ese punto de partida se trabajaron los tres casos de uso como guías en todo el proceso.

Durante el análisis y el diseño, se identificaron clases entidades del metadato necesarias para estandarizar la solución según la norma ISO 19115, y siguiendo además el patrón de arquitectura por capas se identificaron las clases de negocio y para el acceso a los datos.

Se ha obtenido la implementación de una primera versión del software, en la que todos sus componentes críticamente funcionales han sido construidos como se pueden observar en los diagramas de componentes. Se puede concluir que en esta primera versión del software se ha logrado interpretar correctamente las necesidades del cliente y las especificidades del perfil cubano para metadatos.

## CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

### Introducción

Durante la fase de inicio en un proyecto de desarrollo de software se realiza la concepción de todo el problema a resolver y se decide si seguir con el proyecto o no. Esto se logra mediante correctas técnicas de planificación y estimación, que tiene en cuenta las funcionalidades principales que se quieren informatizar, y sus niveles de dificultad. Estas estimaciones determinan el costo que implicaría el desarrollo del proyecto tanto de manera económica, organizativa, técnica y además los beneficios que se obtendrían con su realización.

### 5.1 Estimación basada en el análisis de los Puntos de Casos de Uso.

La estimación mediante el análisis de los Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación de esfuerzo para el desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores de lo afectan, para finalmente contabilizar el tiempo y esfuerzo total estimado para el proyecto.

La asignación de “pesos” en los Casos de Uso se realiza teniendo en cuenta la complejidad o la cantidad de transacciones realizadas con los actores que no son mas que pares de pasos acción-actor/respuesta-sistema que definan actividades relevantes dentro de un caso de uso. El análisis del factor de peso de los actores se realiza teniendo en cuenta si es un usuario común que se comunica mediante interfaces gráficas o un sistema informático que lo hace mediante interfaces de programación o canales de comunicación de algún tipo, y las cantidades de actores que existan en un sistema dado.

#### 5.1.1 Calcular los Puntos de Caso de Uso sin ajustar (UUCP).

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor se calcula a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Donde,

- **UUCP:** Puntos de Casos de Uso sin ajustar.
- **UAW:** Factor de Peso de los Actores sin ajustar.
- **UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

Se calcula el Factor de Peso de los Actores sin ajustar teniendo en cuenta los criterios antes mencionados como se verá en la tabla:

<b>Tipo de actor</b>	<b>Descripción</b>	<b>Factor de peso</b>	<b>Actores</b>	<b>Total</b>
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante una interfaz gráfica.	3	1	3

Tabla 10 Factor de Peso de los Actores sin ajustar.

$$\mathbf{UAW = \sum cant. Actores * peso}$$

$$\mathbf{UAW = 1 * 3}$$

$$\mathbf{UAW = 3}$$

Para calcular el Factor de Peso de los Casos de Uso sin ajustar se tiene en cuenta la cantidad de transacciones se realicen.

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	1	5
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	2	30

Tabla 11 Factor de Peso de los Casos de Uso sin ajustar.

$$\text{UUCW} = \sum \text{cant. Casos uso} * \text{peso}$$

$$\text{UUCW} = 35$$

Finalmente los Puntos de Casos de Uso sin ajustar resultan:

$$\text{UUCP} = 3 + 35$$

$$\text{UUCP} = 38$$

### 5.1.2 Calcular los Puntos de Caso de Uso ajustados (UCP)

Una vez que si tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar el valor mediante la siguiente ecuación:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

Donde,

- **UCP:** Puntos de Casos de Uso ajustados.
- **UUCP:** Puntos de Casos de Uso sin ajustar.
- **TCF:** Factor de Complejidad Técnica.
- **EF:** Factor Ambiente.

El coeficiente de Factor de Complejidad Técnica se calcula mediante la cuantificación de un conjunto de valores que determinan la complejidad técnica del sistema. Cada uno de estos factores se cuantifica con un valor de 0 a 5 donde 0 significa irrelevante y 5 muy importante.

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor asignado</b>	<b>Total</b>
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	2	2
T4	Funcionamiento Interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0,5	5	2.5
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	2	2
T11	Incluye objetivos especiales de seguridad	1	0	0
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	2	2

Tabla 12 Factor de Complejidad Técnica.

$$\text{TCF} = 0,6 + 0,01 * \sum(\text{peso} * \text{valor asignado})$$

$$\text{TCF} = 0,6 + 0,01 * 34$$

$$\text{TCF} = 0,6 + 0,34$$

$$\text{TCF} = 0,94$$

El Factor Ambiente recoge las habilidades y el entrenamiento del equipo de trabajo con el que se planea desarrollar el proyecto. Los valores del aporte se establecen de 0 a 5 según el criterio anterior.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4.5
E2	Experiencia en la aplicación	0,5	4	2
E3	Experiencia en la orientación a objetos.	1	5	5
E4	Capacidad del analista líder.	0,5	4	2
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part – Time	-1	5	-5
E8	Dificultad del lenguaje de programación	-1	1	-1

Tabla 13 Factor Ambiente.

$$EF = 1,4 - 0,03 * \sum(\text{peso} * \text{valor asignado})$$

$$EF = 1,4 - 0,03 * 21.5$$

$$EF = 1,4 - 0,645$$

$$EF = 0,75$$

Finalmente, los Puntos de Casos de Uso Ajustados resultan:

$$UCP = UUCP * TCF * EF$$

$$UCP = 38 * 0.94 * 0.75$$

$$UCP = 26.79$$

### 5.1.3 Estimación del esfuerzo a través de los Puntos de Casos de Uso.

Karner originalmente sugirió que cada Punto de Casos de Uso requiere 20 horas-hombre. Posteriormente surgieron otros refinamientos que proponen una granularidad algo más fina, que han sido analizados según este método sin reportar ningún cambio.

El esfuerzo en horas-hombre se calcula de la siguiente forma:

$$E = UCP * CF$$

Donde,

- **E:** Esfuerzo estimado en horas-hombre.
- **UCP:** Puntos de Casos de Uso Ajustados.
- **CF:** Factor de Conversión.

**CF** = 20 Horas-Hombre / Punto de Casos de uso.

$$E = 26.79 * 20$$

$$E = 535.8 \text{ Horas-Hombre.}$$

Calculando el esfuerzo de todo el proyecto:

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	133.95
Diseño	20	267.9
Implementación	40	535.8
Pruebas	15	200.925
Sobrecarga (otras actividades)	15	200.925
Total	100	1339.5

Tabla 14 Esfuerzo del Proyecto

Si  $E_T = 1339.5$  horas-hombre y cada mes los desarrolladores trabajan como promedio 168 horas en el mes, estoy haría un:

$E_T = 7.97$ meses-hombre.
----------------------------

#### 5.1.4 Costo del Proyecto.

Se estima que el salario promedio mensual sea de \$100.00

**CHM = 2 \* Salario Promedio**

**CHM = 200.00 \$/mes**

**Costo = Salario Promedio\* ET**

**Costo = 100 \* 7.97**

**Costo= \$ 797.00**

## **5.2 Beneficios tangibles e intangibles**

La instalación y explotación del software en las diferentes oficinas de la ONRM donde existan bases de datos geológicas, permitirá la creación de los metadatos de cada una de las bases de datos por todo el país y al tener toda esta información centralizada, llegará de manera más eficiente teniendo en cuenta la agilidad con que la toma de decisiones de la actualidad lo exige.

Por otra parte, el software desarrollado ha seguido los estándares para la gestión de metadatos geológicos, esto garantiza un entendimiento a nivel internacional entre los datos de nuestras aplicaciones y otras interesadas en todo el mundo, de manera que podamos brindar nuestra información de manera rápida para la comercialización y estudio a nivel internacional de los recursos geológicos cubanos.

## **5.4 Análisis de costos y beneficios**

El desarrollo del software no constituye un gasto económicamente significativo, puesto que las herramientas que han sido usadas son libres, el software no tiene ningún requisito de software o hardware que dificulte su instalación, la ONRM posee toda la tecnología necesaria para usarlo.

El costo del proceso de desarrollo es muy bajo como se ha visto, teniendo en cuenta que el software ha sido diseñado e implementado por estudiantes en la universidad, contribuyendo así al proceso de formación de los futuros ingenieros.

## **5.5 Conclusiones**

En este capítulo se ha estimado el tiempo de duración del desarrollo del proyecto, usando las técnicas de Karner basadas en el análisis de Puntos de Casos de Uso. Se ha calculado además el esfuerzo en horas hombres y posible costo de producción del software que no excede los 800 pesos. Teniendo en cuenta las facilidades que brinda el uso del producto, y el bajo costo productivo se decide que el proyecto es factible y se recomienda su desarrollo completo.

## CONCLUSIONES

Durante el desarrollo de este trabajo de diploma se ha realizado un estudio de las técnicas de capturas de metadatos geológicos y los estándares utilizados internacionalmente, y se decidió que es altamente beneficioso desarrollar una solución informática para el perfil de metadatos cubano que cumpla los requisitos de los clientes, y le añada valor al mismo de manera que se convierta en una herramienta de eficaz y de agrado por parte de los usuarios.

Llegado el documento ha servido para la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación haciendo uso de la metodología RUP.

Se alcanzó satisfactoriamente el objetivo propuesto: diseñar una aplicación de escritorio flexible y de fácil entendimiento que proporcione soporte a los procesos de gestión de metadatos geológicos en la ONRM. Reafirmando así la utilidad y validez de emplear las tecnologías informáticas para apoyar las labores que se desarrollan en cualquier tipo de esfera. Se obtuvieron además los siguientes resultados:

1. Se ha demostrado la eficacia de los lenguajes y tecnologías utilizadas para el diseño del sistema.
2. Se realizó el diseño y modelado de la estructura de datos para el Metadato geológico propuesto por el perfil cubano, la cual estará soportada sobre archivos XML donde se almacenarán estos datos, asegurando la validez y calidad de los mismos.
3. Se realizó el análisis y diseño del sistema.
4. La solución propuesta ha sido acertada, los requerimientos soportan al sistema y los casos de uso satisfacen las necesidades funcionales.

Las tareas de la investigación fueron cumplidas, haciendo uso de los métodos científicos planteados garantizando la solidez y profundidad de la investigación.

Fue de vital importancia la utilización de la metodología RUP, que sirvió de guía durante todo el proceso de realización del software, haciendo uso para llevar a cabo el mismo, de la herramienta CASE Visual Paradigm, ahorrando mucho tiempo en la programación gracias a sus eficientes facilidades de generación de código.

Los tres módulos desarrollados que forman parte del sistema de metadatos que contendrá todo el catálogo nacional de metadatos geológicos cubano, permitirán una agilización

considerable de actividades y decisiones de la ONRM, que se verá reflejada considerablemente en la economía nacional.

En conclusión el software ha utilizado tecnologías libres que permiten un costo mínimo en cuestiones de licencias, ha sido desarrollado hasta ahora por estudiantes de la UCI, siguiendo las normas de calidad y administración de proyectos que ofrece la universidad, poseyendo toda la documentación necesaria para su entendimiento, además su costo productivo es muy bajo, por lo que se garantiza su factibilidad.

## RECOMENDACIONES

Durante el levantamiento de requisitos con el cliente, en los que se definieron tres módulos informáticos que realizarían el trabajo de la captura y gestión de los metadatos, la aplicación para la creación de los mismos cuenta con la misión fundamental de crear y actualizar metadatos. A petición del usuario, solo trabajaría sobre ficheros XML para evitar instalaciones de gestores de bases de datos que dificultarían el trabajo de especialistas no entrenados en estas tareas. Sin embargo, durante el estudio de otras soluciones se pudo ver que los software que realizaban el mismo negocio en otras partes del mundo usaban un gestor de base de datos para el almacenamiento de los metadatos y para luego su posterior exportación a diferentes servidores y gestores de información.

Esto es considerablemente una ventaja, puesto que los ficheros XML, son de fácil edición, y aún siendo generados en la aplicación podrían ser modificados fuera de ella y los datos se verían comprometidos, perdiendo su legitimidad. De la misma manera estos ficheros pudieran extraviarse y perder así la información, teniendo que realizar el trabajo nuevamente. Por tanto, se recomienda potencialmente, para futuras versiones del software, un gestor de bases de datos que almacene todos los metadatos, que sea capaz de importarlos y exportarlos en ficheros XML, y facilite su edición y manipulación.

Se recomienda además la revisión del modelo de entidades persistentes del metadato realizado en este trabajo para la implementación de un modelo físico de base de datos, y teniendo en cuenta que, según la recomendación anterior, la futura base de datos deberá importar metadatos en formato XML, se recomienda estudiar y aplicar, según las características del software, las especificidades de la ISO 19139 para esquemas XML sobre datos geológicos, esto permitirá un entendimiento mayor entre todas las aplicaciones de este tipo, y elevará la calidad y seguridad de los metadatos geológicos cubanos.

Durante la implementación del software se obtuvo un prototipo no funcional con el objetivo de responder a las necesidades inmediatas de los clientes pero que no cuenta con las características de interfaz que han sido propuestas, y tampoco con la función de ayuda antes definida. Por tanto, se recomienda también desarrollar estas características para futuras versiones para elevar la calidad y agrado de la aplicación elevando así la satisfacción de los clientes.

## REFERENCIAS BIBLIOGRÁFICAS

1. apuntes.rincondelvago.com. **Apuntes Universitarios . Apuntes de Geología.** [En línea] [http://apuntes.rincondelvago.com/apuntes\\_universidad/geologia/](http://apuntes.rincondelvago.com/apuntes_universidad/geologia/).
2. estrucplan.com.ar. **Estruc Plan On Line. Enciclopedia Geológica Básica - Geología - Definición, relación con otras ciencias.** [En línea] <http://www.estrucplan.com.ar/producciones/entrega.asp?identrega=368>.
3. GeoDato, Programa. **GeoDatoIC. Documentación Técnica.** [Documento Word] La Habana : Geominera Oriente, Diciembre 2005.
4. José A. Senso, Antonio de la Rosa Piñeiro. **El concepto de Metadato. Algo más que descripción de recursos electrónicos.** [PDF] España. : Universidad de Granada., 2003.
5. GeoDato, Programa. **Procedimiento para la Documentación del Metadato.** [Documento Word] La Habana : Geominera Oriente, Octubre 2007.
6. Carbajal, Juan Salcedo. **Informe Final sobre Metadatos ISO 19115.** Lima, Peru : Instituto Geologico Minero y Metalurgico INGEMET, 2007.
7. Sturm, Jack. **Desarrollo de Soluciones XML.** [PDF] [ed.] McGraw Hill Interamericana. s.l. : Microsoft, 2003.
8. autores, Colectivo de. **Estructura para la Base de Datos de Metadatos.** [PDF] La Habana : Instituto de Geología y Paleotología, 2004.
9. ISO. **Geographic Information Metadata. ISO/FDS 19115.** s.l. : International Standard, 2003.
10. María A. Mendoza Sánchez. **Metodologías De Desarrollo De Software.** 2004. [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
11. Terry Quatrani. **Visual Modeling with Rational Rose 2000 and UML.** *Publisher* Addison Wesley. 1999
12. Colectivo de profesores de Ingeniería de Software. **Conferencia UML. 2005**
13. Craig Larman. **UML y Patrones. Introducción al análisis y diseño orientado a Objetos y al proceso unificado.** Prentice Hall.
14. Boch Grady, J.I., Rumbaugh James, **El proceso unificado de desarrollo de software. 20**

## BIBLIOGRAFÍA

1. apuntes.rincondelvago.com. **Apuntes Universitarios . Apuntes de Geología.** [En línea] [http://apuntes.rincondelvago.com/apuntes\\_universidad/geologia/](http://apuntes.rincondelvago.com/apuntes_universidad/geologia/).
2. estrucplan.com.ar. **Estruc Plan On Line. Enciclopedia Geológica Básica - Geología - Definición, relación con otras ciencias.** [En línea] <http://www.estrucplan.com.ar/producciones/entrega.asp?identrega=368>.
3. GeoDato, Programa. **GeoDatoIC. Documentación Técnica.** [Documento Word] La Habana : Geominera Oriente, Diciembre 2005.
4. José A. Senso, Antonio de la Rosa Piñeiro. **El concepto de Metadato. Algo más que descripción de recursos electrónicos.** [PDF] España. : Universidad de Granada., 2003.
5. GeoDato, Programa. **Procedimiento para la Documentación del Metadato.** [Documento Word] La Habana : Geominera Oriente, Octubre 2007.
6. Carbajal, Juan Salcedo. **Informe Final sobre Metadatos ISO 19115.** Lima, Peru : Instituto Geologico Minero y Metalurgico INGEMET, 2007.
7. Sturm, Jack. **Desarrollo de Soluciones XML.** [PDF] [ed.] McGraw Hill Interamericana. s.l. : Microsoft, 2003.
8. autores, Colectivo de. **Estructura para la Base de Datos de Metadatos.** [PDF] La Habana : Instituto de Geología y Paleotología, 2004.
9. ISO. **Geographic Information Metadata. ISO/FDS 19115.** s.l. : International Standard, 2003.
10. María A. Mendoza Sánchez. **Metodologías De Desarrollo De Software.** 2004. [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
11. Terry Quatrani. **Visual Modeling with Rational Rose 2000 and UML.** *Publisher* Addison Wesley. 1999
12. Colectivo de profesores de Ingeniería de Software. **Conferencia UML. 2005**
13. Craig Larman. **UML y Patrones.** *Introducción al análisis y diseño orientado a Objetos y al proceso unificado.* Prentice Hall.

14. Boch Grady, J.I., Rumbaugh James, **El proceso unificado de desarrollo de software.**  
**2000**
15. Hernández León, Rolando Alfredo y Coello González, Sayda. **EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA. s.l. : EDUNIV, 2002.**
16. MINBAS. **Resolución 215 Clasificación de los recursos y reservas de minerales sólidos.**

## ANEXOS

### **Anexo 1.** Modelo de Entrevistas con el cliente.

Pregunta 1: ¿Qué son los metadatos?

Pregunta 2: ¿Quiénes los posibles clientes interesados en hacer búsquedas en el Metadato?

Pregunta 3: ¿La información del Metadato estará disponible para todos los usuarios de internet?

Pregunta 4: ¿La propuesta de un Especialista, y su rol para entrar datos online, es correcta?

Pregunta 5: Al pensar en un sistema se han montado todos los requisitos funcionales en casos de uso del sistema, pero no se conocen los procedimientos del ministerio al definir nuevos clientes o posibles clientes quienes verán toda la información del Metadato. ¿Cómo funcionan legalmente todos estos trámites?

Pregunta 6: ¿Cómo se decide quien ve la información en el Metadato?

## Anexo 2. Diagramas de secuencias de Análisis.

### Caso de uso: Crear Metadato.

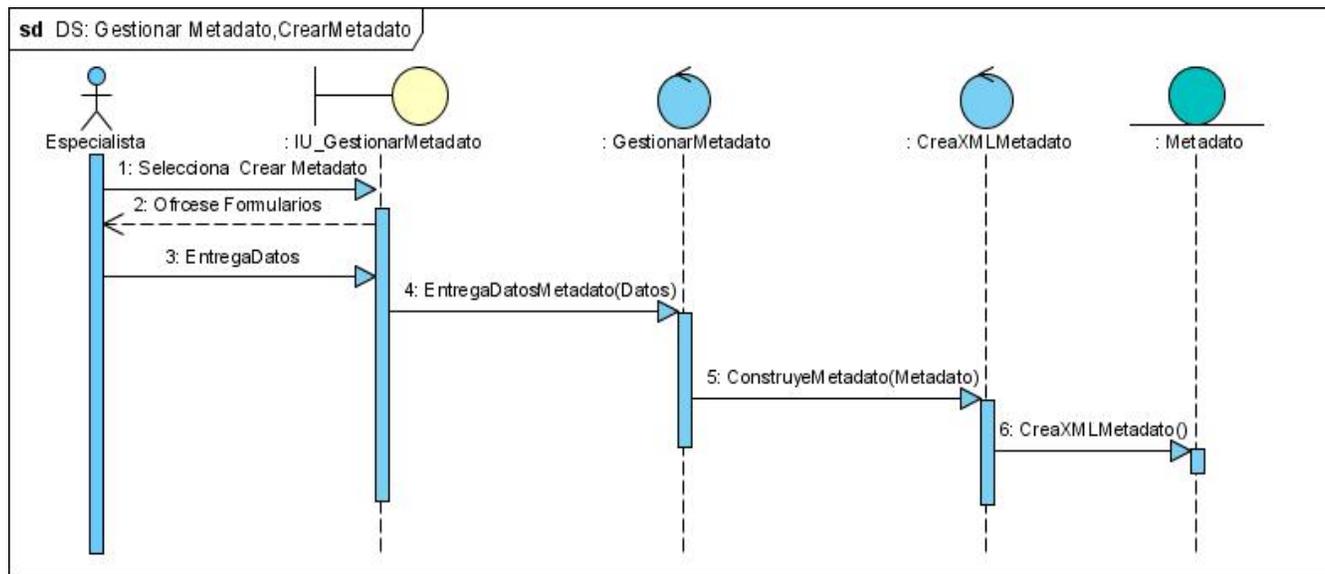


Fig. A.2.1 Diagrama de Secuencia: Capturar Metadato

### Caso de Uso: Actualizar Metadato

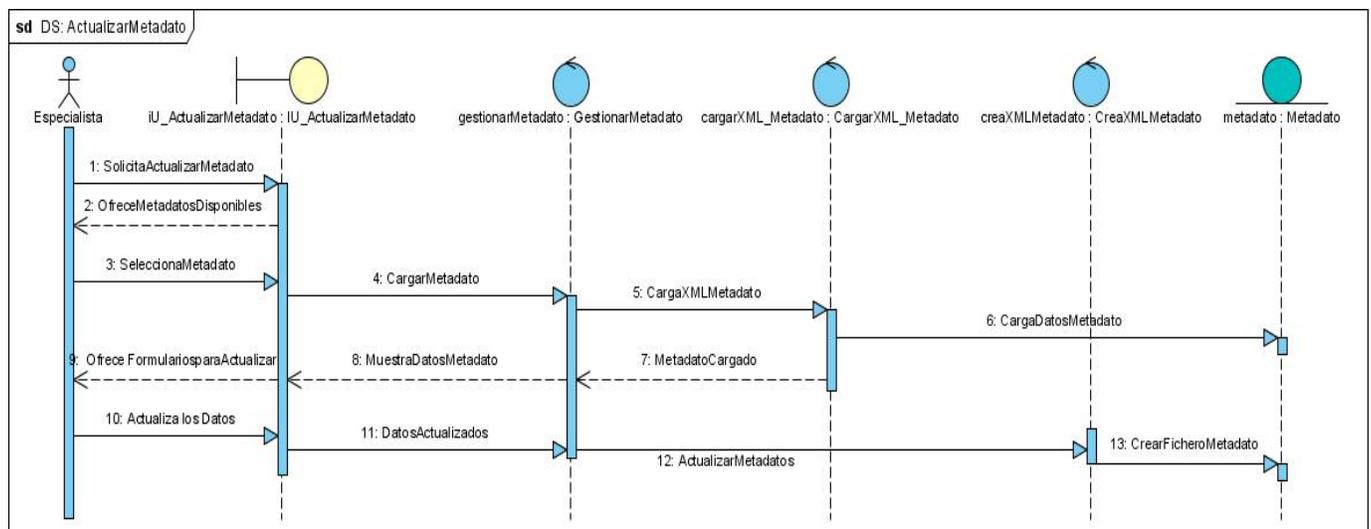


Fig. A.2.2 Diagrama de Secuencia: Actualizar Metadato.

## Caso de uso: Actualizar Nomencladores

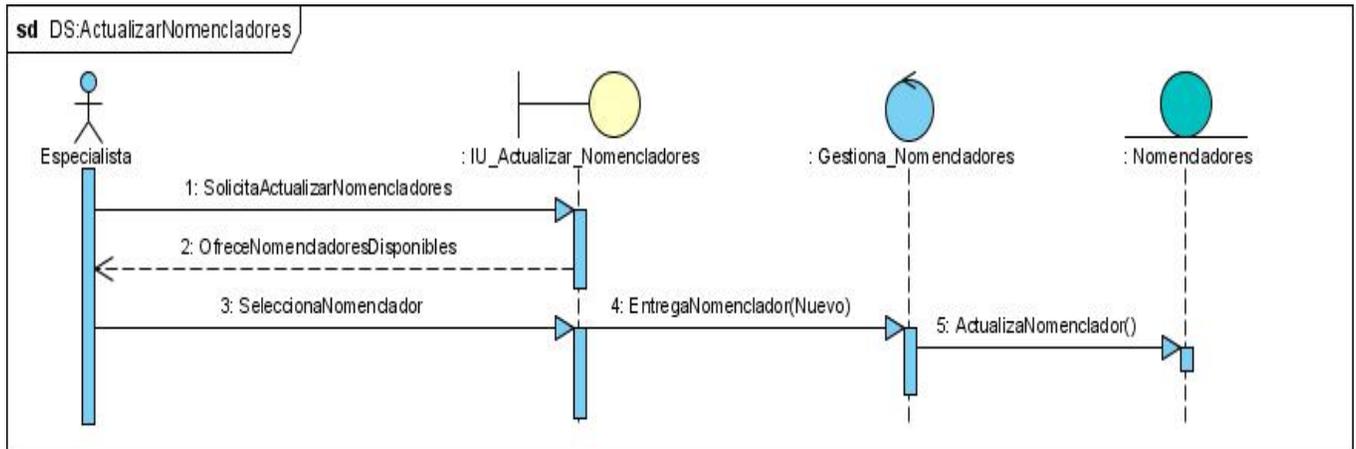


Fig. A.2.3 Diagrama de secuencia: Actualizar Nomencladores.

### Anexo 3. Diagrama de clases de diseño de interfaz.

#### Clases Interfaz para Paquete: Información del Metadato.

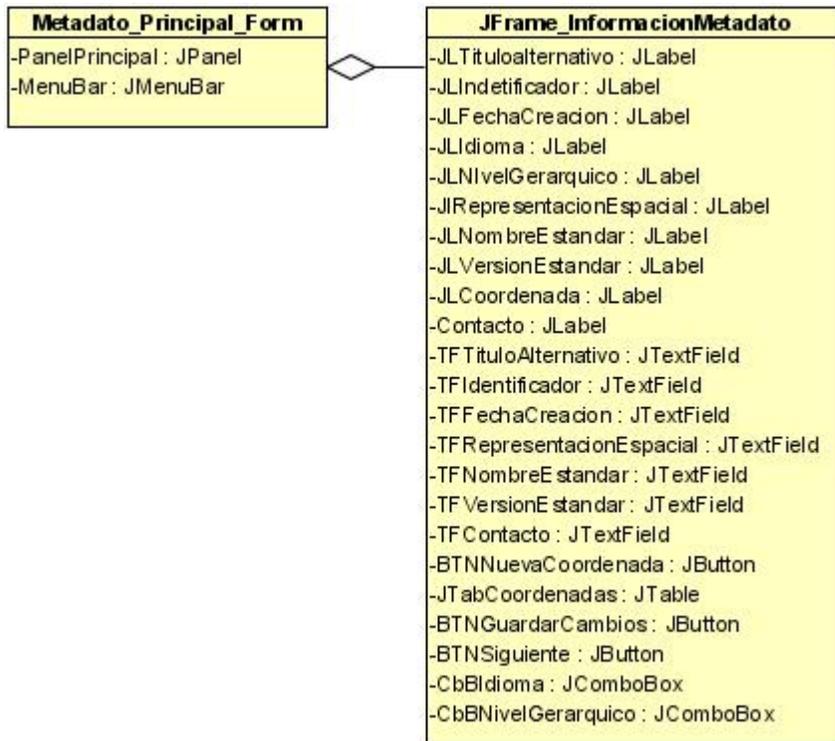


Fig. A.2.1 Diagrama de clase de interfaz: Información Metadato.

## Clases Interfaz para Paquete: Identificación de Datos

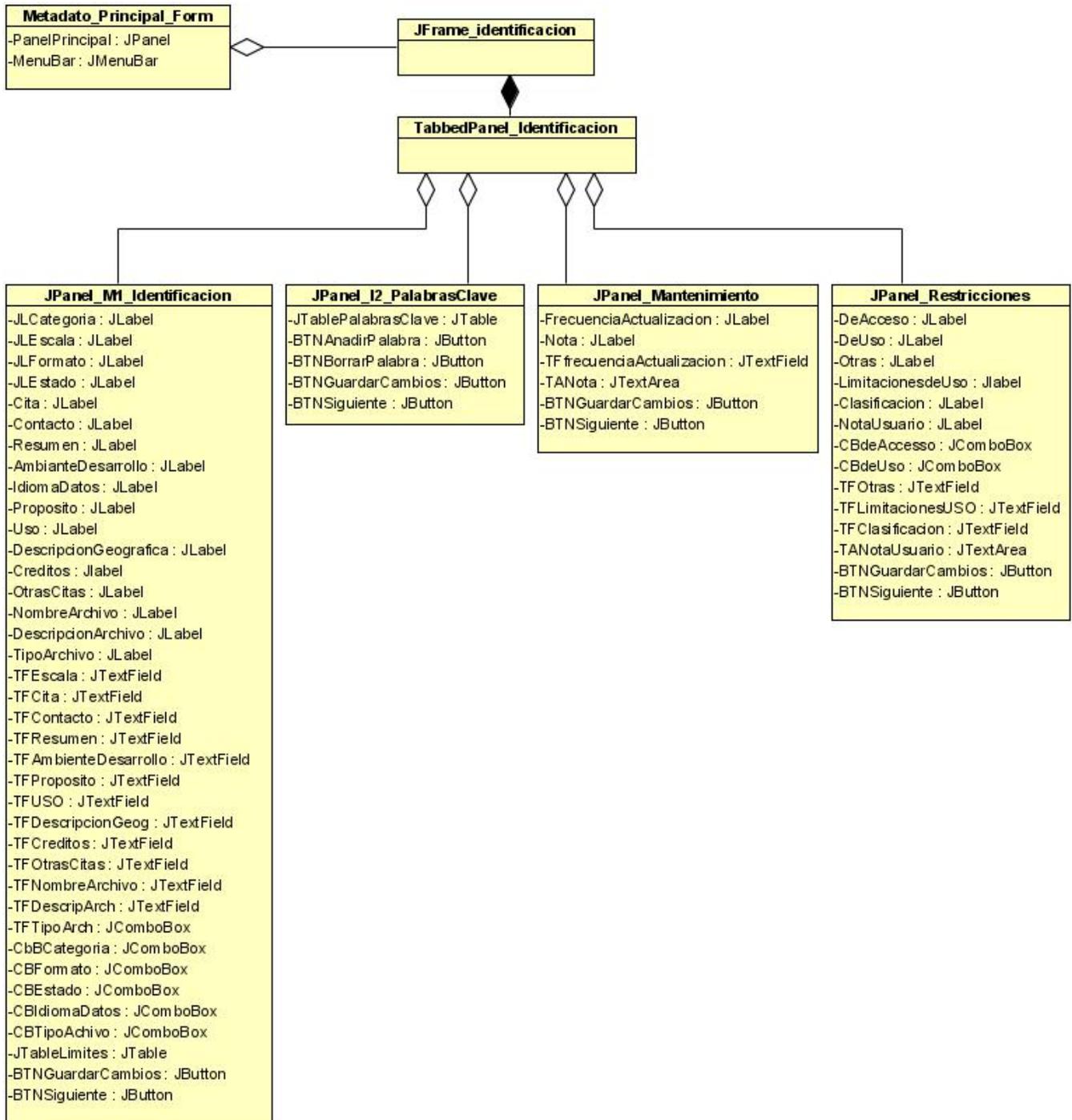


Fig. A.3.2 Diagrama de clase de interfaz: Identificación de los Datos.

### Clases Interfaz para Paquete: Información del Contenido.

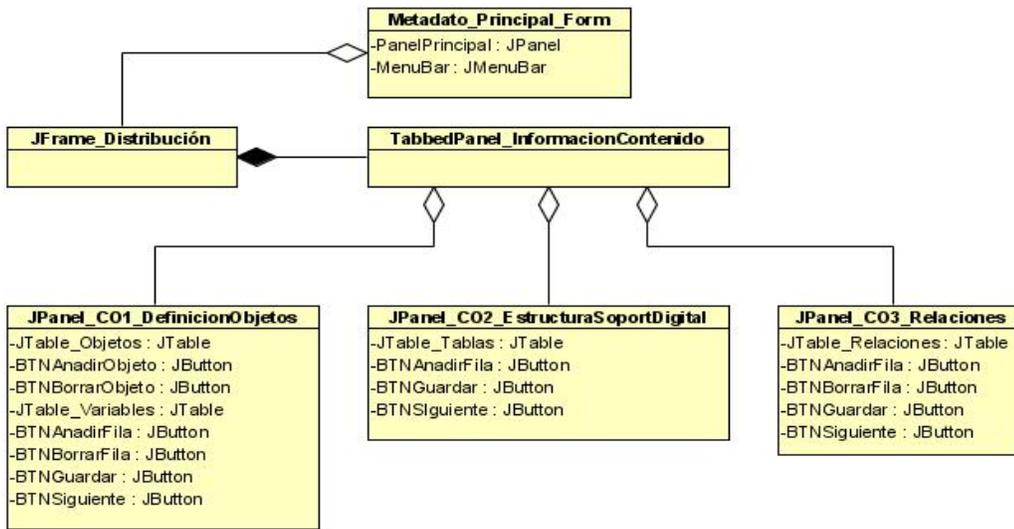


Fig. A.3.3 Diagrama de clase de interfaz: Información del Contenido.

### Clases Interfaz para Paquete: Calidad de los Datos.

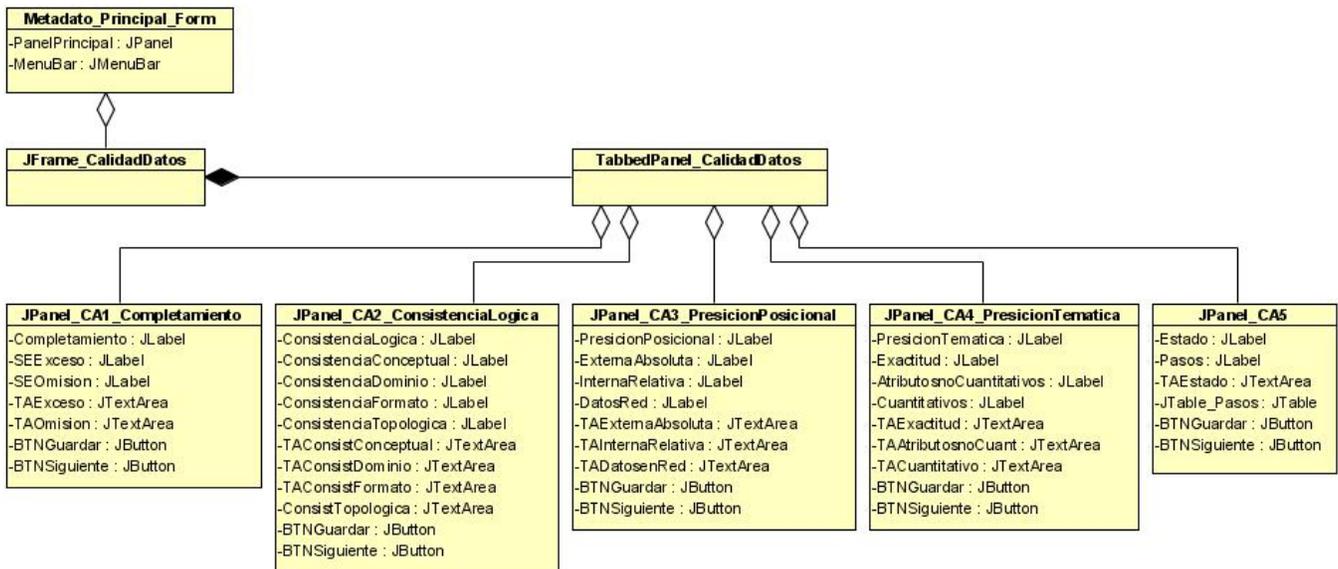


Fig. A.3.4 Diagrama de clase de interfaz: Calidad de los Datos.

### Clases Interfaz para Paquete: Representación Espacial.

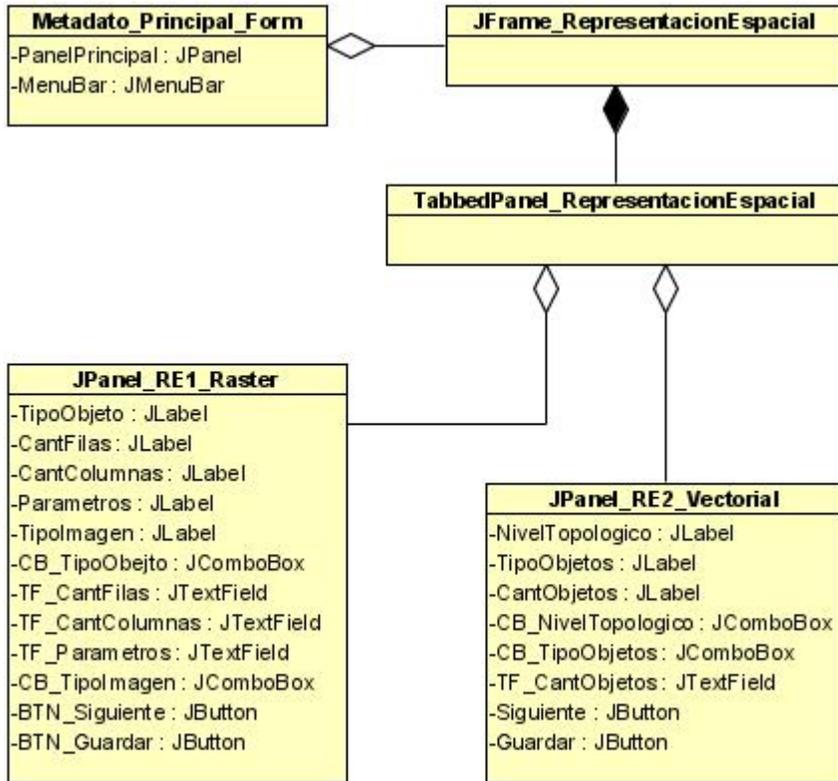


Fig. A.3.5 Diagrama de clase de interfaz: Representación Espacial.

### Clases Interfaz para Paquete: Sistema de Referencia.

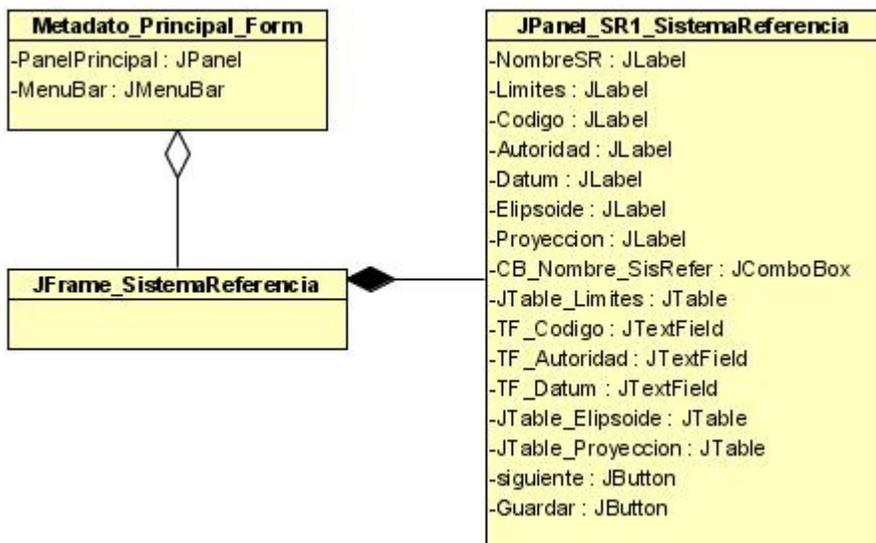


Fig. A.3.6 Diagrama de clase de interfaz: Sistema de Referencia

### Clases Interfaz para Paquete: Distribución de los Datos

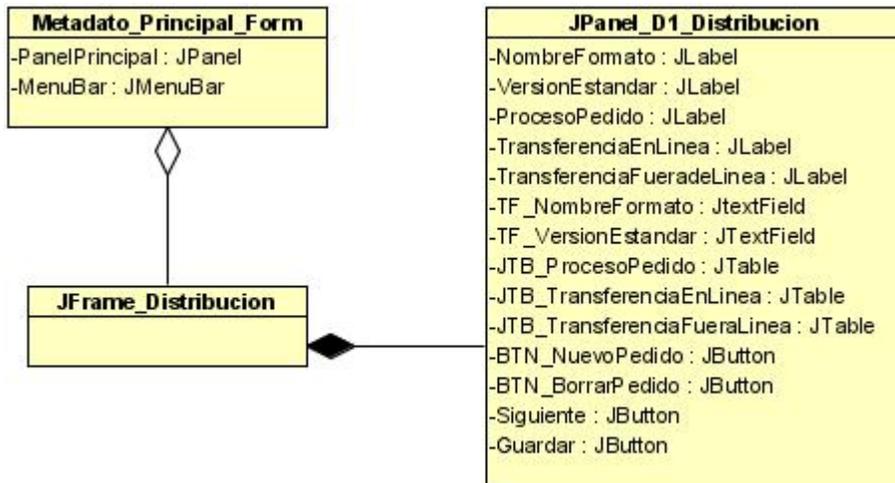


Fig. A.3.7 Diagrama de clase de interfaz: Distribución de los Datos.

## Clases interfaz para Paquete: Información sobre Citas

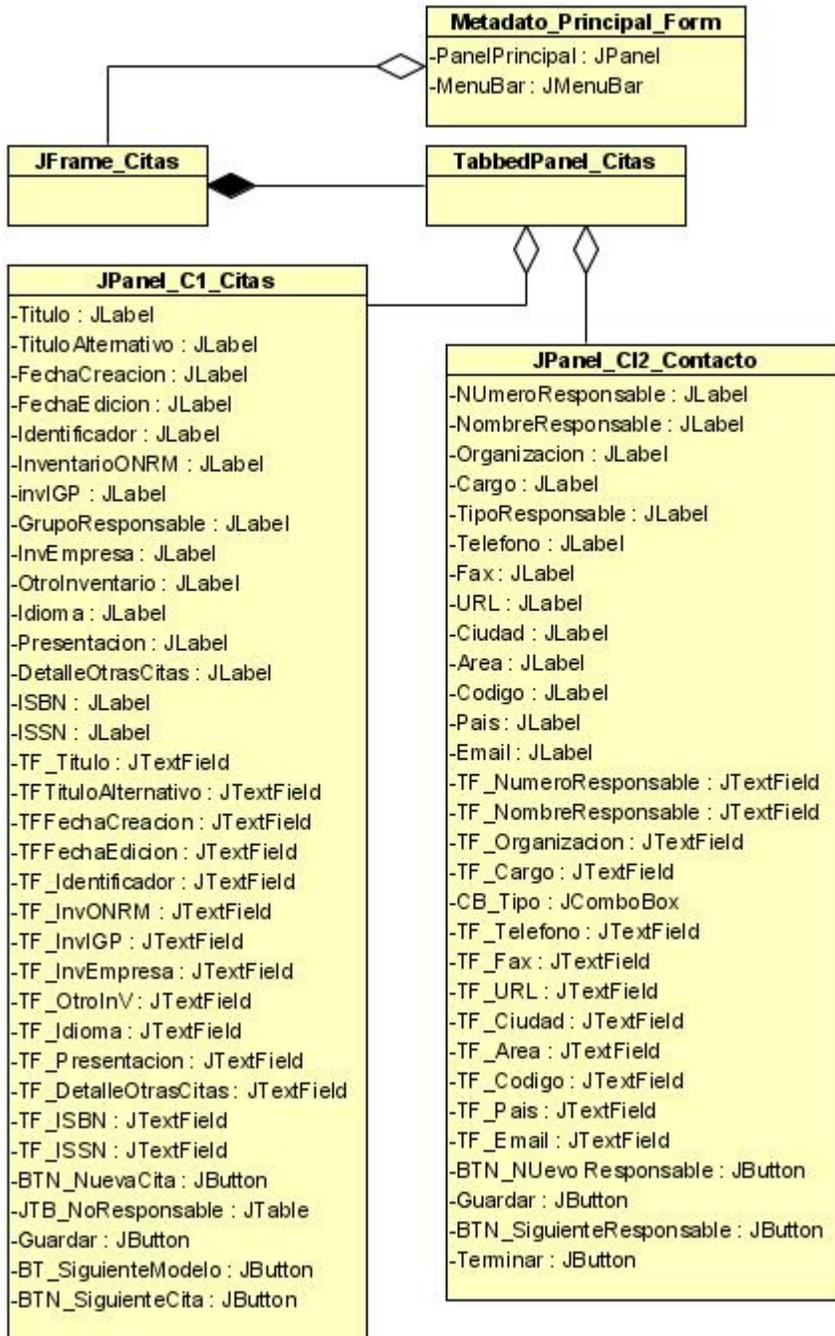


Fig. A.3.8 Diagrama de clase de interfaz: Información sobre Citas.

## Anexo 4. Prototipos de interfaz.

The screenshot shows a software window titled "Llenar Modelos del Paquete Información del Metadato". The main content area is labeled "Metadato. Modelo M-1 Datos Generales del Conjunto de Datos (obligatorio)". It contains several input fields and dropdown menus:

- Título Alternativo:
- Identificador:
- Fecha de Creación:
- Nombre del Estándar:
- Versión del Estándar:
- Fecha del Estándar:
- Idioma:
- Nivel Jerárquico:
- Representación Espacial:

Below these fields is a section for "Coordenadas" with fields for "Vértice X", "Vértice Y", "Vértice Z", and "Sistema de Coordenadas" (set to "Cuba Norte"). A "+" button is next to the system field.

At the bottom, there are two buttons: "Siguiete Paquete" and "Guardar Metadato".

Fig. A.4.1 Vista Prototipo (Paquete Información del Metadato).

The screenshot shows a software window titled "Llenar Modelos del Paquete Identificación de los Datos". It has a tabbed interface with four tabs: "Modelo I-1. Identificación", "Modelo I-2 Palabras Claves", "Modelo I-3 Mantenimiento", and "Modelo I-4 Restricciones". The active tab is "Modelo I-1. Identificación (obligatorio)".

The main content area includes the following fields and controls:

- Categorías o Temas:
- Escala:
- Formato:
- Estado:
- Cita:
- Contacto:
- Idioma:

Below these are several text areas with scrollbars and arrows, labeled "Ambiente de Desarrollo", "Propósito", "Uso", "Descripción Geográfica", "Resumen", "Otras Citas", and "Créditos".

The "Archivo" section contains:

- Nombre del Archivo:
- Tipo de Archivo:
- Descripción de Archivo:

At the bottom, there is a "Límites" section with fields for "Norte", "Sur", "Este", and "Oeste", followed by a "+" button. Below this are three buttons: "Anterior Paquete", "Siguiete Paquete", and "Guardar Metadato".

Fig. A.4.2 Vista Prototipo (Paquete Identificación de los Datos).

The screenshot shows a software window titled "Información del Contenido". It has three tabs: "Modelo CO-1 Definición de Objetos" (selected), "Modelo CO-2 Estructura de Base de Datos", and "Modelo CO-3 Relaciones". Below the tabs, the text "Metadato. Modelo CO-1 Definición de Objetos (obligatorio)" is displayed. The main area contains two sections. The first section has a header "Objeto" and "Descripción" and three rows of input fields: "Rasgo", "Tipo de Rasgo", and "Perfil de Trabajo". The second section has a header "Descripción" and two rows of input fields: "Variable Virtual" and "Variable Elemental", with a "+" button to the right of the "Variable Elemental" field. At the bottom, there are three buttons: "Adicionar Objeto", "Anterior Paquete", "Siguiete Paquete", and "Guardar Metadato".

Fig. A.4.3 Vista Prototipo (Paquete Información del Contenido).

The screenshot shows the same software window, but with the "Modelo RE-1 Representación Espacial Raster" tab selected. The text "Metadato. Representación Espacial. RE-1 Raster" is displayed. The main area contains two rows of input fields. The first row has "Tipo de Objeto" (a dropdown menu showing "Matrix"), "Cantidad de Filas" (a text input field), and "Cantidad de Columnas" (a text input field). The second row has "Parametros" (a text input field with up and down arrow buttons) and "Tipo de Imagen" (a dropdown menu showing "Radar"). At the bottom, there are three buttons: "Anterior Paquete", "Siguiete Paquete", and "Guardar Metadato".

Fig. A.4.4 Vista Prototipo (Paquete Calidad de los Datos)

Modelo RE-1 Representación Espacial Raster    Modelo RE-2 Representación Espacial Vectorial

Metadato, Representación Espacial, RE-1 Raster

Tipo de Objeto:     Cantidad de Filas:     Cantidad de Columnas:

Parametros:     Tipo de Imagen:

Anterior Paquete    Siguiete Paquete    Guardar Metadato

Fig. A.4.5 Vista Prototipo (Paquete Representación Espacial).

Modelo SR-1 Sistema de Referencia

Metadato, Modelo SR-1, Sistema de Referencia

Nombre del Sistema de Referencia:     Código:     Autoridad:

Límite:    Norte:     Sur:     Este:     Oeste:

Elipsoide	Proyección
Semi Eje Menor: <input type="text"/>	Paralelo Estándar: <input type="text"/>
Semi Eje Mayor: <input type="text"/>	Longitud Meridiano Central: <input type="text"/>
Unidades: <input type="text"/>	Longitud Origen Proyección: <input type="text"/>
Achatamiento: <input type="text"/>	Falso Este: <input type="text"/>
	Falso Norte: <input type="text"/>
	Unidad: <input type="text"/>
	Factor Escala: <input type="text"/>

Anterior Paquete    Siguiete Paquete    Guardar Metadato

Fig. A.4.6 Vista Prototipo (Paquete Sistema de Referencia).

Metadato. Modelo D-1 Distribución de los Datos

Formato  Version

Procesos de Pedido

Tarifa  Instrucciones  Entrega  Contacto

Tarifa	Instrucciones	Entrega	Contacto

Adicionar

Eliminar

Transferencia

Unidad  Tamaño

En Línea

Enlace

Función

Fuera de Línea

Nombre del Medio

Densidad

Formato

Anterior Paquete    Siguiente Paquete    Guardar Metadato

Fig. A.4.7 Vista Prototipo (Paquete Distribución de los Datos).

LLenar Modelos del Paquete Información Sobre Citas

Modelo CI-1 Información Sobre Citas    Modelo CI-2. Grupo de Responsables

Metadato.Modelo CI-1 Información Sobre Citas.(obligatorio)

Título Alternativo

Identificador

Título

Fecha de Edición

Fecha de Creación

Inventario ONRM  Inventario IGP  Inventario Empresa

Otro Inventario  Idioma  Presentación

ISBN  ISSN  No. Responsables  +

Detalles de Otras Citas

Adicionar Cita

Anterior Paquete    Guardar Metadato

Fig. A.4.8 Vista Prototipo (Paquete Información sobre Citas).

## Anexo 5. Paquetes de Implementación del Metadato.

### Paquete: Información del Contenido

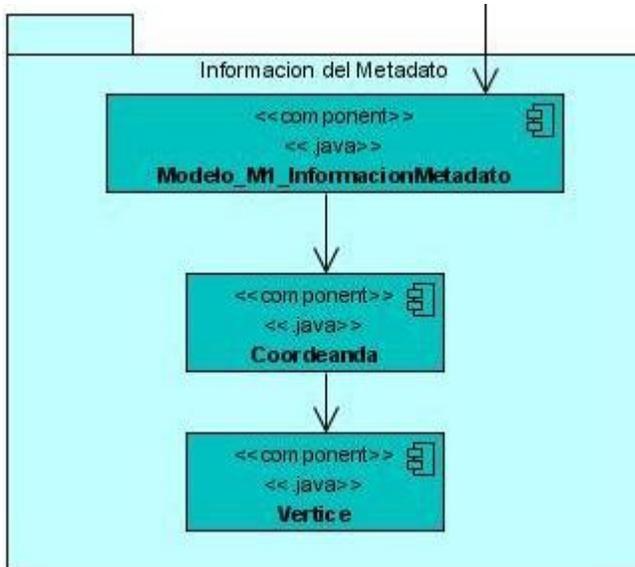


Fig. A.5.1 Diagrama de componentes: Información del contenido.

### Paquete: Identificación de los Datos

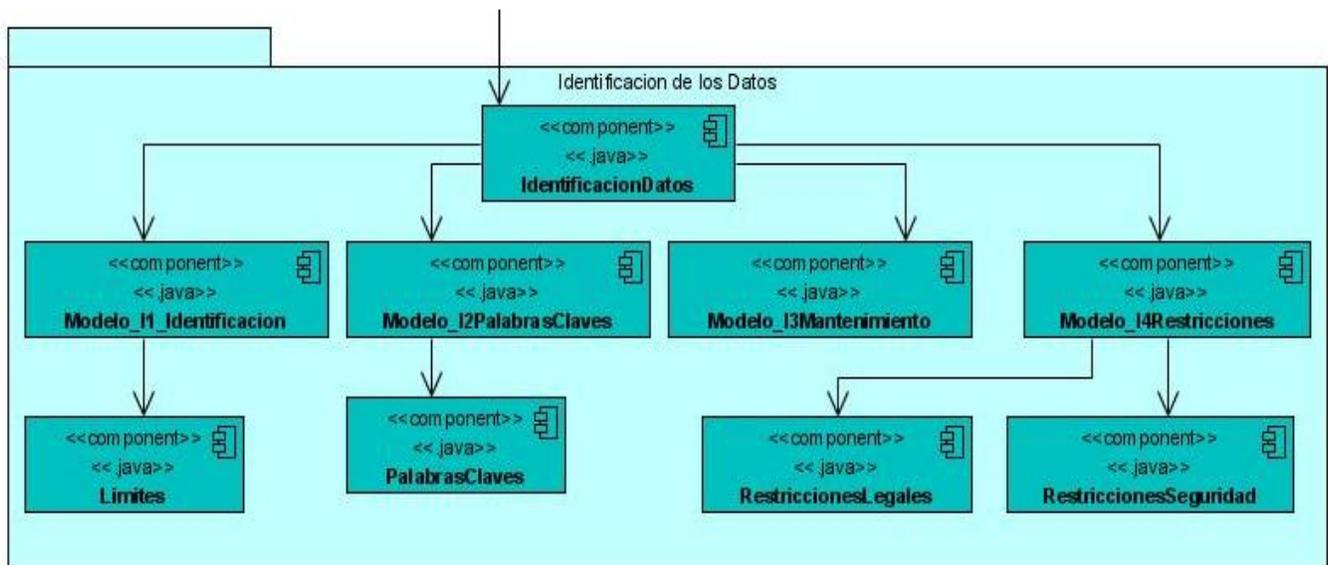


Fig. A.5.2 Diagrama de componentes: Identificación de los Datos.

### Paquete: Información del Contenido

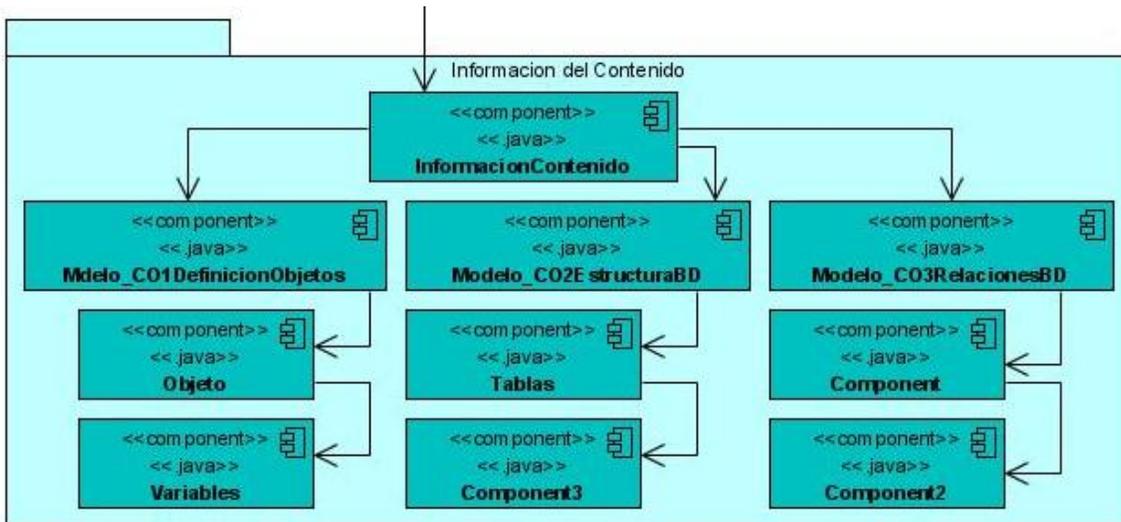


Fig. A.5.3 Diagrama de componentes: Información del Contenido.

### Paquete: Calidad de los Datos.

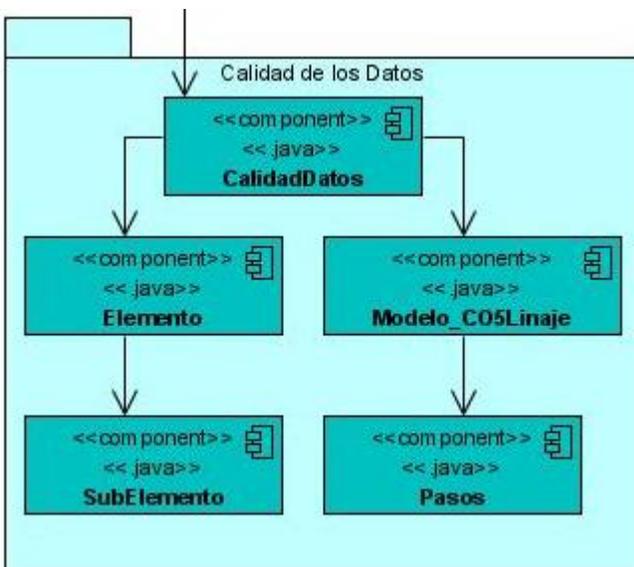


Fig. A.5.4 Diagrama de componentes: Calidad de los Datos.

### Paquete: Representación Espacial

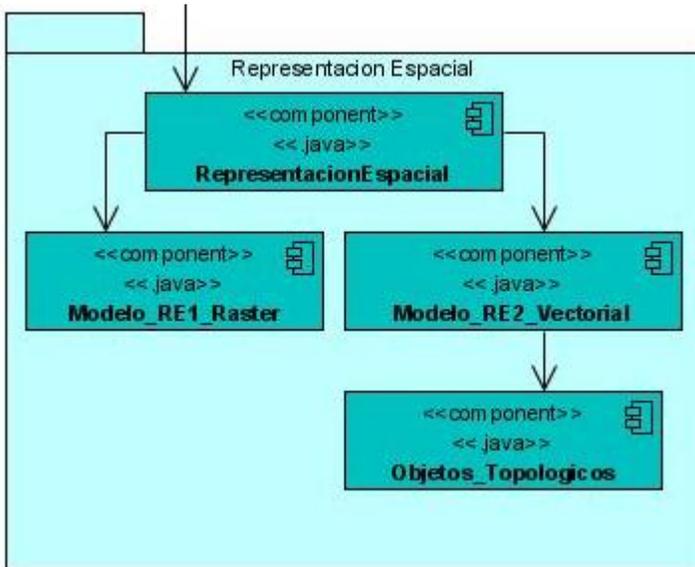


Fig. A.5.5 Diagrama de componentes: Representación Espacial.

### Paquete: Sistema de Referencia

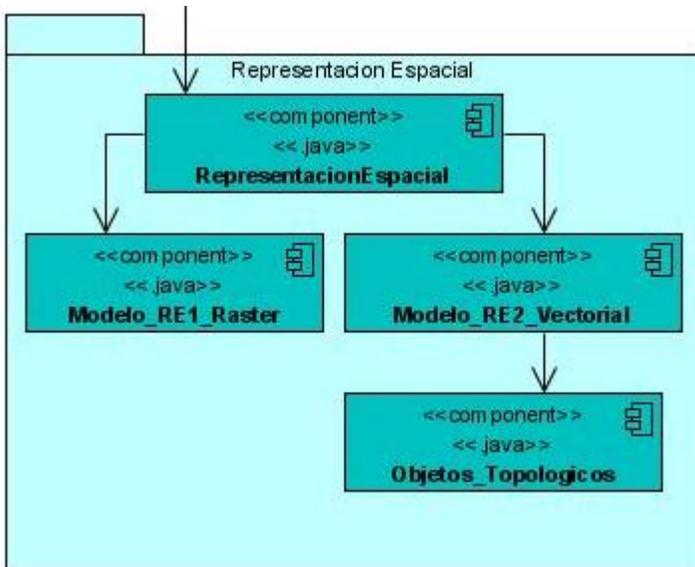


Fig. A.5.6 Diagrama de componentes: Sistema de Referencia

### Paquete: Distribución de los Datos

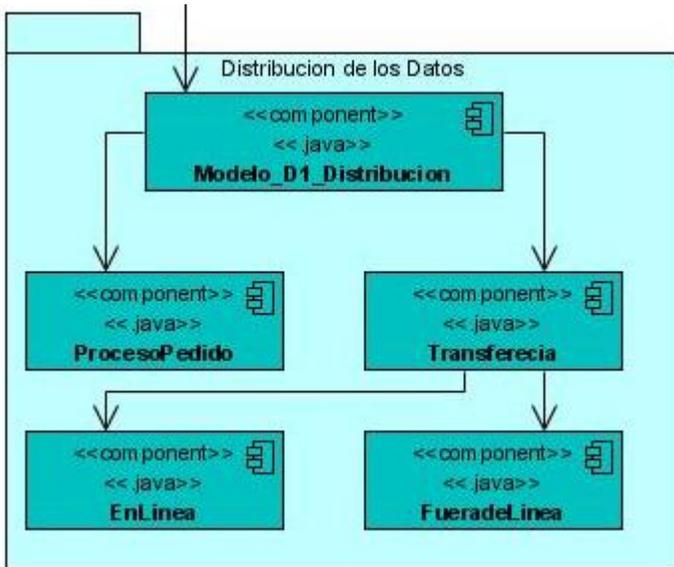


Fig. A.5.7 Diagrama de componentes: Distribución de los Datos.

### Paquete: Información sobre Citas.

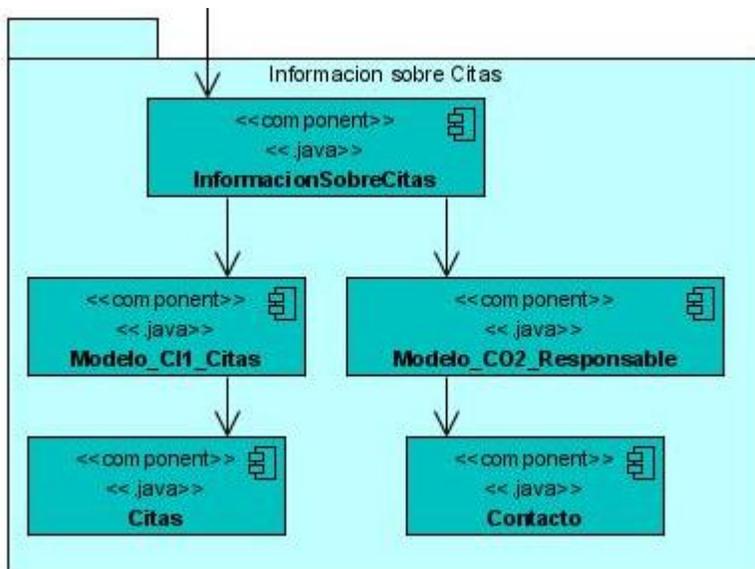


Fig. A.5.8 Diagrama de componentes: Información sobre Citas

## Anexo 6: Paquete de implementación Interfaz Usuario.

### Paquete: Interfaz Usuario

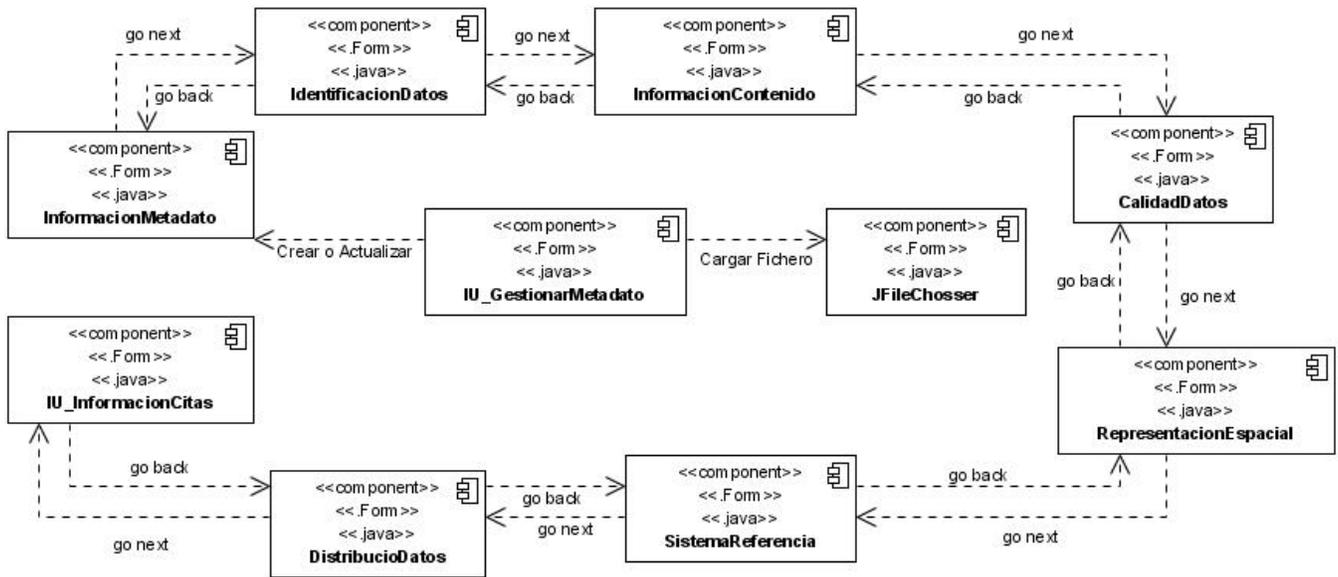


Fig. A.6.1 Diagrama de componentes

## GLOSARIO DE TÉRMINOS

API: Del inglés *Application Programming Interface* - Interfaz de Programación de Aplicaciones; es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Applet: Componente de una aplicación que se ejecuta en el contexto de otro programa.

ArcCatalog: Software libre que se encarga de la captura y edición de los metadatos geológicos siguiendo el estándar ISO 19115.

BDARV: Tecnología de Base de Datos con Atributos y Registro de Variables. Técnica utilizada en las oficinas de la ONRM para la edición de las base de datos geológicas.

C/C++: Lenguajes de Programación.

CASE: *Computer Aided Software Engineering* - Ingeniería de Software Asistida por Ordenador- son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

CatMDEdit: Software libre destinado a la gestión de los metadatos geológicos siguiendo las normas ISO 19115.

Especialista: Persona entrenada en la gestión de los metadatos geológicos cubanos.

GeoDato: Programa de la ONRM para la informatización del Conocimiento Geológico Cubano.

Geología: Es una ciencia natural que describe la composición del planeta, su ubicación y actuación en el universo.

Geólogo: Persona responsable de una base de datos geológicos.

ICSW: Industria Cubana de Software.

IDE: Del inglés *Integrated Development Environment*- Entorno de Desarrollo Integrado.

INGEMMET: Perfil de metadatos desarrollado en Lima, Perú.

JAVA: Lenguaje de programación de alto nivel, orientado a objetos.

Linux: Sistema Operativo desarrollado sobre la política de software libre.

MetaD: Aplicación desarrollada en España para la gestión de los metadatos geológicos.

Metadato: Datos sobre otros datos.

MSF: de inglés *Microsoft Solution Framework*, es una metodología usada para el desarrollo de software.

ONRM: Oficina Nacional de Recursos Minerales.

RUP: Del inglés *Rational Unified Process*- Proceso Unificado de Rational, es una metodología usada para el desarrollo de software.

Ubuntu: Nombre de una distribución del sistema operativo Linux.

UML: Del inglés *Unified Modeling Language*, Lenguaje Unificado de Modelado.

XML: Del inglés *Extensible Markup Language*, Lenguaje extensible por etiquetas.

XP: Del inglés *Extreme Programming*, metodología de desarrollo de software.