

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



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

Herramienta para la creación de Metadatos Geológicos.

AUTOR: Alain Leon Companioni

TUTOR: Ing. Carlos Luis Serrano Rosales.

Ciudad de La Habana, julio del 2010.

Año 52 de la Revolución.

Dedicatoria

*... a mis abuelos Ana y Guillermo, que aunque ya no estén, siempre los recuerdo y seguro
estarían muy orgullosos.*

*... a mis padres quienes confiaron plenamente en mí desde el principio, les dedico mi título que
más que mío es de ellos.*

... a mi abuela Rosa y a su fiel compañero Tata, por brindarme su amor y su apoyo.

... a mi hermana, quien ha sido una gran amiga y de quien siempre quise seguir sus pasos.

Agradecimientos

- ... especialmente a mis padres, quienes son mi razón de ser y me apoyaron incondicionalmente y son una gran inspiración para mí.
- ... a mi hermanita, chiquita de tamaño, pero grande de corazón, mujer que se enfrenta a los problemas que la vida le pone en su camino de manera extraordinaria.
- ... a mis abuelos que siempre estuvieron ahí para aconsejarme y guiarme por buenos caminos.
- ... a Lázaro, por tener tanta fe en mí, espero no defraudarlo y llegar a ser la persona que quiere que sea.
- ... a mi tutor Carlos Luis por guiarme en todo momento y estar siempre dispuesto cuando lo necesité.
- ... a mis amigos del IPVCE, especialmente Reisbel y Yoani, los cuales vienen luchando conmigo todos estos años y espero que muchos más.
- ... a la gente del Brood, Tony, Eyonys y Adrian, compañeros inseparables de noches interminables en el laboratorio, quienes me enseñaron mucho más que la amistad.
- ... a mis compañeros del antiguo 9104, por estar ahí y compartir momentos de alegría y tristeza.
- ... a mis amigos del proyecto, con quienes compartí mucho más de 15 días, donde supimos que lo que era trabajo en equipo y que al final una “Success” no significaba nada por mucho que se lo explicara a Vladimir. A todos, Eddy, Dagoberto, Rocny, Chirino, Figura, Héctor, Yamil, Gerdys, Liester, Vladimir, Ruben, gracias.
- ... a mis compañeros del grupo de CDI, con quienes compartí mucho más que una ilusión y el sueño de montar en un avión.
- ... a los pesaítos del 26 Reinaldo, Alberto, Raúl, Joel, Rolando.
- ... a la gente de la PMM Amed, Alexander, Eddy, Gustavo
- ... a todos los que de una forma u otra me ayudaron, no solo a que esta investigación se hiciera posible, también a los que me han ayudado a llegar donde estoy hoy.

Declaración de autoría

Declaración de Autoría.

Declaro que soy el único autor de este trabajo y autorizo a 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 _____.

Alain Leon Companioni

Ing. Carlos Luis Serrano Rosales

Datos de contacto

Datos de Contacto.

TUTOR:

Nombre: Ing. Carlos Luis Serrano Rosales.

Correo Electrónico: cserrano@uci.cu

Universidad de las Ciencias Informáticas.

Resumen

Hoy en día, la generalización del concepto "metadatos" da cobertura a gran parte de la información que se maneja en el mundo, tal es el caso de la geología, ciencia que estudia todo lo referente a la composición, estructura, formación y otras características de la tierra, y los elementos que la componen. A partir de la necesidad de organizar toda la información generada por este estudio, surge el concepto de metadato geológico.

En Cuba una de las entidades autorizadas para la generación, administración y uso fundamental del conocimiento geológico es la Oficina Nacional de Recursos Minerales (ONRM). Dicha entidad, perteneciente al Ministerio de la Industria Básica (MINBAS), constituye la autoridad minera en el país.

El elevado número de datos e información que se maneja en la ONRM, aparejado a la no existencia de una herramienta que permita la captura de los metadatos geológicos como parte de esta información, genera la necesidad de una aplicación informática que cumpla con este propósito, a partir del uso del estándar cubano definido para el trabajo con metadatos geológicos en el país. Dicho estándar tiene su fundamento en la norma ISO 19115, permitiendo a su vez la creación de archivos XML a partir de la especificación técnica ISO 19139.

Se presenta el resultado del ciclo completo de desarrollo de software que propone la metodología RUP, mostrando como resultado una aplicación de escritorio que permite agilizar los procesos de creación de metadatos geológicos en la ONRM.

Palabras Claves:

Metadatos, Geología, Metadato Geológico, ISO 19115, ISO 19139, XML, Aplicación de escritorio.

Datos en Inglés

Nowdays, the generalization of the term "metadata" covers much of the information handled in the world, such is the case of geology, science of everything related to the composition, structure, training and other features land, and the elements that compose it. Since the need to organize all the information generated by this study, does the concept of geological metadata.

In Cuba one of the authorized entities for generation, management and basic use of geological knowledge is the National Office of Mineral Resources (ONRM). This entity, under the Ministry of Basic Industry (MINBAS), is the mining authority in the country

The high number of data and information used in the ONRM, coupled with the nonexistence of a tool that allows the capture of metadata as part of this geologic information, generates the need for a software application that meets this purpose, from the use of the Cuban standard defined for geological work with metadata in the country. This standard is based on ISO 19115, while allowing the creation of XML files based on the technical specification ISO 19139.

The investigation presents the results of the complete cycle of software development methodology proposed RUP, displaying results in a desktop application that allows you to streamline the process of creating metadata in the ONRM geological.

Keywords

Metadata, Geology, Geological Metadata, ISO 19115, ISO 19139, XML, Desktop Application.

Índice

Introducción	1
Capítulo 1:	5
1.1. Introducción	5
1.2. Conceptos asociados al dominio del problema.	5
1.2.1 Geología.....	5
1.2.2 Datos Geológicos.....	6
1.2.3 Metadatos.	6
1.2.4 Metadatos geológicos.	7
1.3 Objeto de estudio.	8
1.3.1 Descripción general.	8
1.3.2 Descripción actual del dominio del problema.	11
1.3.3 Situación problemática.	13
1.4 Otras soluciones existentes.	13
1.4.1 GeoNetwork.	13
1.4.2 CatMDEdit.....	14
1.4.3 MetaD	16
1.4.4 Comparación de las soluciones existentes.	16
Capítulo 2:	18
2.1 Introducción.	18
2.2 Las Tecnologías de la Información y las Comunicaciones.	18
2.3 Arquitectura de software.	19
2.3.1 Arquitectura Modelo-Vista-Controlador	20
2.4 Lenguaje Unificado de modelado.	20
2.5 Metodologías de Desarrollo de Software.	21
2.5.1 RUP	22
2.5.2 XP	22
2.5.3 MSF.	23
2.5.4 Selección de la metodología a utilizar.	24
2.6 Lenguajes de programación	25
2.6.1 Java.	25
2.6.2 C#	27
2.6.3 C++	28

Índice

2.6.4	Selección del lenguaje de programación a utilizar	29
2.7	XML.....	29
	Ventajas de XML:	30
2.7.1	Justificación del lenguaje XML.....	30
2.8	Herramientas CASE.....	31
2.8.1	Visual Paradigm.....	31
2.8.2	Rational Rose.....	32
2.8.3	Selección de la Herramienta CASE a utilizar.....	32
2.9	Entornos de desarrollo.....	33
2.9.1	NetBeans.....	33
2.9.2	Eclipse.....	34
2.9.3	Selección del entorno de desarrollo a utilizar.....	34
2.10	Framework.....	35
2.10.1	Framework a utilizar: Swing.....	35
2.11	Sistema Gestor de Base de Datos.....	36
2.11.1	Sistema gestor de base de datos a utilizar: HSQLDB.....	36
2.12	Conclusiones parciales.....	37
Capítulo 3:	38
3.1	Introducción.....	38
3.2	Modelo de Negocio.....	38
3.2.1	Actores y trabajadores del negocio.....	38
3.2.2	Procesos de negocio.....	40
3.2.3	Diagrama de Casos de Uso del Negocio.....	40
3.2.4	Descripción textual de los Casos de Uso del Negocio.....	41
3.2.4.1	Descripción Textual del Caso de Uso: Consultar Metadato.....	41
3.2.4.2	Descripción Textual del Caso de Uso: Archivar Metadato.....	42
1.1.1.1	Descripción Textual del Caso de Uso: Revisar Metadato.....	43
3.3	Requisitos funcionales.....	44
3.4	Requisitos no funcionales.....	44
3.5	Descripción del sistema propuesto.....	45
3.1.1	Descripción de los actores del sistema.....	45
3.1.2	Diagrama de casos de usos del sistema.....	46
3.1.3	Descripción de los casos de usos del sistema.....	46

Índice

3.6 Conclusiones parciales.....	48
Capítulo 4:	49
4.1 Introducción.....	49
4.2 Modelo de diseño.	49
4.2.1 Patrones utilizados durante el diseño.	49
4.2.2 Diagrama de clases del diseño del CU Gestionar datos de responsables.....	50
4.3 Principios de diseño.....	50
4.3.1 Estándares de interfaz de la aplicación.....	50
4.3.2 Concepción general de la ayuda.....	51
4.4 Diseño de la Base de Datos.....	51
4.4.1 Diagrama de clases persistentes.	52
4.4.2 Diagrama entidad relación.	53
4.5 Diagrama de Despliegue.....	54
4.6 Diagrama de Componentes.....	54
4.7 Pruebas al sistema propuesto.....	55
4.7.1 Caso de uso Gestionar Datos de Responsables.....	55
4.8 Conclusiones parciales.....	60
Conclusiones Generales.....	61
Recomendaciones.....	62
Referencias bibliográficas	63
Bibliografía.....	65
Glosario de términos.....	67

Tablas y Figuras

Índice de Tablas.

Tabla 1 Comparación de las soluciones existentes.....	16
Tabla 2: Descripción de los actores del negocio.	39
Tabla 3: Descripción de los trabajadores del negocio.	39
Tabla 4: Descripción Textual del Caso de Uso: Consultar Metadato.....	42
Tabla 5: Descripción Textual del Caso de Uso: Archivar Metadato.....	42
Tabla 6: Descripción Textual del Caso de Uso: Revisar Metadato.....	44
Tabla 7: Descripción de los actores del sistema.	46
Tabla 8 Descripción textual del CU cargar configuración.	48
Tabla 9 Descripción textual de CU Gestionar datos de responsables.	48
Tabla 10 Secciones y escenarios CU Gestionar Datos de Responsables.....	56
Tabla 11 Descripción de las variables CU Gestionar Datos de Responsables.....	57
Tabla 12 Matriz de datos Sección Adicionar Responsable.....	59

Índice de Figuras.

Figura 1: Metadato. Secciones y modelos de documentación.....	11
Figura 2: Diagrama de Casos de Usos del Negocio.....	40
Figura 3: Diagrama de casos de usos del sistema.	46
Figura 4: Diseño CU Gestionar datos de responsables.....	50
Figura 5 Diagrama de clases persistentes.	52
Figura 6 Diagrama entidad relación.	53
Figura 7 Diagrama de Despliegue.....	54
Figura 8 Diagrama de componentes del CU Gestionar datos de Responsable.....	54

Introducción

La propia evolución de la humanidad está estrechamente relacionada con la habilidad que tienen los seres humanos de adaptarse al medio ambiente que los rodea, todo esto, gracias a la capacidad que tiene el cerebro humano de aprender. Tanto la conciencia social como la del hombre en particular se han convertido, con el paso del tiempo, en un centro de acumulación de información. La sociedad humana se ha convertido en un complejo sistema informativo, cognitivo y transformador.

La base de toda información la constituyen los datos que, por sí solos, carecen de sentido, es decir no tienen ningún valor semántico, pero si éstos son procesados correctamente y organizados se puede generar la información necesaria para producir el conocimiento.

Con el transcurso del tiempo, la humanidad se vio en la necesidad de organizar de alguna forma toda la información acumulada por generaciones pasadas, en libros, documentos y diversos formatos, pero se hacía cada vez más difícil la búsqueda sobre algún tema, debido a la enorme cantidad existente. Es por ello que en las bibliotecas de tiempos pasados aparecieron los catálogos de libros, los cuales agilizaban el proceso de búsqueda de información. Con el surgimiento de los catálogos de libros se crearían las bases para una nueva forma de organizar los datos mismos, los metadatos.

Hoy en día la generalización del concepto de metadatos ha cubierto cualquier tipo de información, tal es el caso de la geología, ciencia que estudia todo lo referente a la composición, estructura, formación, entre otras características, de la tierra y los elementos que la componen, lo que hace necesario tener un modo de organizar toda la información generada por este estudio. **(1)**

En Cuba, el estudio de las ciencias geológicas ha tenido un crecimiento considerable en los últimos años, convirtiéndose en uno de los sectores más importantes para el país.

El Ministerio de la Industria Básica (MINBAS) es la organización estatal cubana que tiene la responsabilidad de tres importantes sectores de la economía de nuestro país, la Energía, la Geología y la Minería así como la Química Básica. Una de los objetivos fundamentales de esta organización, es la investigación geológica, búsqueda, explotación y procesamiento de minerales sólidos, extracción, producción y comercialización mayorista de minerales sólidos, sal y sus derivados, investigación geológica y búsqueda de aguas minero medicinal. **(2)**

Una de las entidades autorizadas para la generación, administración y uso fundamental del conocimiento geológico es la Oficina Nacional de Recursos Minerales (en lo adelante ONRM). Esta entidad es una de

Introducción

las empresas pertenecientes al MINBAS y constituye la autoridad minera en el país, fue constituida en el año 1995 mediante la Ley de Minas. Tiene como principales funciones el control y fiscalización del Proceso Concesionario en Cuba, la protección de los recursos minerales y de hidrocarburos, el control de la explotación racional de los recursos minerales y la preservación del medio ambiente durante el desarrollo de actividades mineras. **(3)**

El elevado número de datos e información que se maneja en la ONRM, que hacen referencia al desarrollo geológico del país, en su mayoría archivados en papel o en varias bases de datos con diversos formatos, provoca atrasos en la localización y procesamiento de información, pérdida de documentos, así como la repetición y errores en la información geológica manejada.

Las anteriores dificultades entorpecen el desarrollo del país y el de la propia entidad, un ejemplo lo constituyen las empresas, tanto nacionales como extranjeras, que desean establecer relaciones en la esfera geológica en el país, a éstas, se les hace muy difícil este proceso, pues no tienen un fácil acceso a la información geológica.

El vertiginoso desarrollo de las Tecnologías de la Información y las Comunicaciones (en lo adelante TIC), posibilita un rápido acceso a grandes fuentes de información, gran capacidad de almacenamiento, así como la digitalización de la información.

Con todos los elementos anteriores, surge el Programa Nacional de Informatización del Conocimiento Geológico (en lo adelante PNICG) como un resultado del trabajo conjunto de varias entidades geológicas nacionales, la ONRM y la Universidad de las Ciencias Informáticas (en lo adelante UCI).

Para dar cumplimiento al acuerdo de trabajo, en la mencionada Universidad se crea, dentro de las instancias productivas del Polo Geoinformática perteneciente a la Facultad 9 el proyecto “Sistema de Gestión de Datos Geológicos” con el objetivo de Desarrollar aplicaciones para la ONRM que permitan manipular de forma segura la información que se genera, manipula o transfiere en la mencionada entidad cubana.

Un área de trabajo del mencionado proyecto persigue garantizar la perdurabilidad de la información del conocimiento geológico del país, incrementar la velocidad de localización de los datos geológicos, mejorar el procesamiento de los datos dentro de la oficina o por sus clientes, ofrecer mejor eficiencia en cuanto a la accesibilidad de la información geológica, mantener un control adecuado sobre el acceso a la información asegurando su integridad. Esta requiere de una solución inmediata para evitar el deterioro de

Introducción

la información, logrando formatos estándares en su almacenamiento y una digitalización adecuada para conformar un Metadato a través de los datos geológicos.

Por lo todo lo anteriormente expuesto, se puede identificar el siguiente **problema a resolver**: ¿Cómo agilizar el proceso de creación de Metadatos Geológicos que forma parte del Sistema de Gestión de Datos Geológicos?

Definiendo como **objeto de estudio** los procesos de creación de Metadatos Geológicos, donde el **campo de acción** lo constituye la informatización de los procesos de creación de Metadatos Geológicos en la Oficina Nacional de Recursos Minerales, teniéndose como **objetivo general** desarrollar una aplicación que permita la creación de Metadatos Geológicos que forma parte del SGD.

Se defiende la siguiente **idea**: el desarrollo de una herramienta informática permitirá agilizar el proceso de creación de Metadatos Geológicos que forma parte del SGD.

Para ello se especificaron las siguientes tareas:

- Caracterizar el proceso de creación de Metadatos en la Oficina Nacional de Recursos Minerales.
- Caracterizar las tendencias y tecnologías actuales a desarrollar.
- Elaborar el Diagrama de Casos de Uso del Negocio del módulo Metadatos.
- Especificar los requisitos funcionales del software.
- Elaborar el Diagrama de Casos de Uso del Sistema del módulo Metadatos.
- Elaborar los diagramas de clase del diseño del módulo Metadatos.
- Elaborar el diagrama de implementación.
- Implementar de los CU definidos.
- Desarrollar los Casos de Prueba que certifiquen la veracidad de los algoritmos empleados.

Los métodos científicos que se utilizarán en el desarrollo de la investigación son los siguientes:

Métodos Teóricos:

- **Análisis y síntesis:** Se usa para el estudio de la bibliografía utilizada y para realizar una recopilación de la misma.
- **Modelación:** Se utiliza para la modelación de diagramas, representar el proceso de desarrollo y propiciar mejor entendimiento de la solución a implementar.

Introducción

- **Análisis histórico - lógico:** Se utiliza para el análisis histórico de los procesos de gestión de información en la ONRM y de las características de los principales servicios geológicos existentes en el mundo.

Métodos Empíricos:

- **Observación:** Se utiliza para identificar los principales procesos que se desarrollan en la ONRM y para definir los requerimientos tanto funcionales como no funcionales.

Al concluir la investigación se esperan los siguientes **resultados:**

1. La Herramienta de creación de Metadatos Geológicos.
2. La documentación técnica asociada al desarrollo de la herramienta de creación de Metadatos Geológicos.

El siguiente trabajo de diploma estará dividido en 4 capítulos fundamentales:

Capítulo 1: Fundamentación Teórica. En este capítulo se expone la descripción del entorno donde se encuentra el negocio y su organización, se describe detalladamente la situación problemática y el análisis de otras soluciones que puedan brindar respuesta al problema científico planteado en el presente trabajo.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar. En este capítulo se explican las principales tecnologías, lenguajes de programación y herramientas que se utilizarán para la construcción de la solución propuesta, así como las ventajas de utilizarla.

Capítulo 3: Presentación de la solución propuesta. En este capítulo se describen los procesos actuales a través de un modelo de negocio, se seleccionan los requisitos funcionales y no funcionales y los casos de uso del sistema, todo esto unido a los correspondientes diagramas que lo modelan.

Capítulo 4: Construcción de la solución propuesta. En este capítulo se plantea la construcción propuesta en el capítulo anterior, en función de diagramas de clases y estándares del diseño, diseño de la base de datos, modelo de despliegue y modelo de implementación.

Capítulo 1: Fundamentación Teórica

Capítulo 1:

Fundamentación Teórica.

1.1. Introducción.

En el presente capítulo se abordan los temas relacionados con la fundamentación teórica de la investigación. Se describen los conceptos asociados al dominio del problema, así como la descripción del objeto de estudio, referente a la creación de metadatos geológicos en la Oficina Nacional de Recursos Minerales, haciendo uso de conceptos y definiciones, además de plantear y profundizar sobre la situación problemática. Se analizan algunas soluciones existentes de gestión en la ONRM y algunas a nivel internacional.

1.2. Conceptos asociados al dominio del problema.

1.2.1 Geología.

La Geología, del griego “geo” (tierra) y “logos” (estudio), es la ciencia que estudia todo lo referente a la evolución, estructura, composición así como otros elementos de la tierra. Esta ciencia tiene varias aplicaciones entre las que se puede encontrar la evaluación de los suelos para la construcción, garantizar un uso correcto de los minerales y aguas subterráneas, entre otras, que hacen de esta ciencia una de las más importantes en los días actuales que vive la humanidad, con los problemas del cambio climático, pues además, analiza los riesgos geológicos en ciertas zonas, como por ejemplo los sismos y la actividad volcánica. **(1)**

Para lograr estudiar todos los elementos del subsuelo, que estudia la Geología, se hizo necesario dividirla en varias ramas, entre las que se pueden encontrar:

- Geología estructural.
- Geología histórica.
- Geoquímica.
- Geofísica.
- Paleontología.
- Mineralogía.
- Hidrogeología.
- Petrología.

Capítulo 1: Fundamentación Teórica

- Sismología.
- Vulcanología.

Cada una de estas ramas de la Geología, realizan investigaciones, generalmente de observaciones en superficies, excavaciones o perforaciones. Toda la información, que generan estas investigaciones, tiene una característica en común, la ubicación espacial, la cual se expresa en coordenadas, así, de esta forma se puede localizar geográficamente el lugar exacto de una investigación, sin importar la rama a la cual pertenece. De esta manera la base y resultados de las investigaciones geológicas están expresados por datos de textos, mediciones discretas, registros continuos o imágenes. **(4)**

1.2.2 Datos Geológicos.

Generalmente localizar un dato geológico que se necesite para un fin determinado se torna muy difícil, pues no existe un mecanismo lo suficientemente general que difunda las características fundamentales de los datos existentes así como su disponibilidad. Se necesita entonces una estructura bien organizada que permita documentar fácilmente la información existente y la que se cree posteriormente, por lo que se decide catalogar los datos haciendo uso de los Metadatos.

Los datos geológicos pretenden modelar y describir el mundo real para su posterior análisis y visualización mediante medios muy diversos. Sus características principales así como sus limitaciones deben estar completamente documentadas mediante los metadatos. **(5)**

1.2.3 Metadatos.

El término metadatos no tiene una definición única. Según la definición más difundida metadatos constituyen “datos sobre datos”. Debido a que muchas veces no se tiene en cuenta la diferencia entre dato e información también hay muchas declaraciones como “informaciones sobre datos”, “datos sobre informaciones” e “informaciones sobre informaciones”. **(6)**

Otras clases de definiciones tratan de precisar el término como “descripciones estructuradas y opcionales que están disponibles de forma pública para ayudar a localizar objetos” o “datos estructurados y codificados que describen características de instancias conteniendo informaciones para ayudar a identificar, descubrir, valorar y administrar las instancias descritas”. Estas definiciones hacen mayor hincapié en los metadatos en relación con la recuperación de información, y surgió de la crítica de que las declaraciones más simples son tan difusas y generales que dificultarán la tarea de acordarse de estándares, pero estas definiciones no son muy comunes.

Capítulo 1: *Fundamentación Teórica*

Los metadatos pueden describir colecciones de objetos y también procesos en los que están involucrados, describiendo cada uno de los eventos, sus componentes y cada una de las restricciones que se les aplican. Definen las relaciones entre los objetos, como las tuplas en una base de datos o clases en orientación a objetos, generando estructuras.

Dependiendo del área donde se empleen, los metadatos tienen varias aplicaciones, pero en general tienen las siguientes ventajas:

1. **Adhieren contenido, contexto y estructura a los objetos de información**, asistiendo de esta forma al proceso de recuperación de conocimiento desde colecciones de objetos.
2. **Permiten generar distintos puntos de vista conceptuales** para sus usuarios o sistemas, y liberan a estos últimos de tener conocimientos avanzados sobre la existencia o características del objeto que describen.
3. **Permiten el intercambio de la información** sin la necesidad de que implique el intercambio de los propios recursos.
4. En cada proceso productivo, o en cada etapa del ciclo de vida de un objeto de información, se van generando metadatos para describirlos y metadatos para describir dichos metadatos (manual o automáticamente). generando de esta forma **valor añadido** a los recursos
5. Permiten un **acceso a los recursos en forma controlada** ya que se conoce con precisión el objeto descrito.
6. Permiten **preservar los objetos de información** permitiendo migrar (gracias a la información estructural) sucesivamente éstos, para su posible uso por parte de las futuras generaciones.
7. Son esenciales para sostener un crecimiento de una Web a mayor escala, permitiendo **búsquedas, integración y recuperación del conocimiento** desde un mayor número de fuentes heterogéneas.

1.2.4 Metadatos geológicos.

La calidad de los datos utilizados es un elemento fundamental en cualquier proceso productivo y no fue hasta hace pocos años que se le empezó a dar la importancia que tiene. La exactitud del dato geográfico, siempre ha estado presente en la creación de datos geológicos pero no se habían adoptado ningún Sistema de Gestión de la Calidad. Uno de los métodos para medir la calidad de los datos geológicos es la utilización de metadatos geológicos. (6)

Capítulo 1: *Fundamentación Teórica*

El uso de metadatos geológicos permite identificar rápidamente al conjunto de datos que satisface un interés determinado. Los metadatos geológicos son imprescindibles en la construcción de una sólida infraestructura de datos geológicos y su utilización facilita la documentación de los datos.

El Comité Federal de Datos Geográficos (2000) reconoce que la creación de metadatos geográficos persigue tres objetivos principales: **(7)**

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

La documentación de los datos se ha convertido en un trabajo necesario para realizar una correcta gestión y explotación de los mismos. El aprovechamiento de los datos no está pensado para el uso interno de una organización o empresa, sino también para poderla compartir e intercambiar entre los diferentes productores de datos, o entre los productores y los usuarios. Para que esta documentación sea comprensible para todos, es decir, para que todos “hablen el mismo idioma”, se han diseñado los estándares de metadatos, en específico, para los metadatos geológicos, que pretenden poner en común los puntos de vista de los diferentes productores de datos, estos estándares son: **(8)**

- La norma ISO 19115.
- La especificación ISO 19139.

1.3 Objeto de estudio.

1.3.1 Descripción general.

Capítulo 1: Fundamentación Teórica

En el mundo de la Información Geográfica se han definido varias recomendaciones para la creación de metadatos geológicos, cuya finalidad principal es proporcionar una estructura “jerárquica y concreta” que permita describir detalladamente cada uno de los datos a los que hacen referencia. Estas recomendaciones han sido creadas y aprobadas por organismos de normalización a partir de opiniones de expertos en esta materia. Estas recomendaciones, en forma de normas o esquemas de metadatos geológicos, suministran criterios para caracterizar sus datos geológicos con propiedad.

La “Organización de Estandarización Internacional (ISO, por sus siglas en inglés)” a través de su familia ISO 19100 define, entre otras temáticas, normas relacionadas con metadatos. Dicha familia se encuentra dividida por comités. En relación con la información geográfica y los metadatos se puede destacar el trabajo del siguiente comité:

- El Comité Técnico 211, denominado “Geomática/Información Geográfica”, ha definido lo siguiente:
 1. La norma “ISO 19115:2003- *Geographic Information Metadata*” que presenta un modelo general de metadatos de información geográfica.
 2. La especificación técnica “ISO/TS 19139-*Geographic Information- Metadata -- XML schema implementation*” para la implementación de los metadatos como XML.
 3. Propuesta de algunas extensiones como “ISO/CD 19115-2 *Geographic information- Metadata-Part 2: Extensions for imagery and gridded data*” para la descripción de recursos de información geográfica específicos.

La norma Internacional ISO 19115:2003- *Geographic Information Metadata* se creó con el objetivo de definir una estructura que sirva para describir los datos geológicos. Esta norma internacional proporciona un modelo y establece un conjunto común de terminologías, definiciones y procedimientos de aplicación para los metadatos geológicos. Mediante la definición de elementos de metadatos se podrá describir la información sobre la identificación, la extensión, la calidad, el modelo espacial y temporal, la referencia espacial y la distribución de los datos geológicos.

Esta Norma de Metadatos es de una gran complejidad e incluye una extensa serie de elementos de metadatos, unos obligatorios y otros opcionales. El documento consta de 140 páginas, incluye un total de 409 ítems y define 27 listas controladas, mediante las que se definen los posibles valores válidos de ciertos campos.

Aunque esta norma define un extenso número de elementos de metadatos, establece un “conjunto mínimo” de datos para conformar un metadato (el núcleo o Core), a considerar para todo el rango de

Capítulo 1: *Fundamentación Teórica*

aplicaciones de los metadatos (desde mapas en formato papel a datos en formato digital, como imágenes satélite, modelos digitales del terreno, etc.). Con este conjunto se pretende establecer un mínimo de parámetros, para facilitar el descubrimiento, el acceso, la transferencia y la utilización de los datos. Este núcleo está formado por elementos obligatorios y otros opcionales que usados todos ellos aumenta la interoperabilidad de los datos y permite a los usuarios entenderlos sin ambigüedades. Todo perfil que se defina a partir de esta norma debe estar formado como mínimo por los elementos definidos en dicho núcleo.

La norma ISO 19115 proporciona una estructura para describir información geográfica mediante elementos de metadatos y establece una terminología común para los mismos pero no desarrolla como poder llevar a cabo su implementación.

La ISO 19139 (*“ISO/TS 19139-Geographic Information-Metadata-XML schema implementation”*) es una especificación técnica que desarrolla una implementación en XML del modelo de metadatos descrito por la ISO 19115, con lo cual se garantiza que los metadatos geológicos tengan una implementación común, la cual pueda ser validada para comprobar que cumple con los requisitos planteados en la norma internacional ISO 19115 para metadatos geológicos.

Como se mencionó anteriormente la ISO 19115 aprueba que aunque el núcleo de los metadatos geológicos es invariable, se pueden hacer variaciones, que contengan al menos el dicho núcleo, es por eso que en Cuba se decide crear una especificación de la norma ISO 19115 para metadatos geológicos, la cual, se ajuste más a las particularidades y necesidades del país.

Es entonces que en el marco del evento GeoDato, después de un análisis del contenido de los metadatos para los conjuntos de datos geológicos, siguiendo la norma ISO 19115 se definen las siguientes secciones para los metadatos geológicos del país: **(5)**

Información del Metadato.

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

Capítulo 1: Fundamentación Teórica

La sección “*Información sobre las Citas*” se ubica al final porque en ella se sitúan las fuentes citadas y los grupos responsables o contactos referidos en el resto de las secciones. Para ello la cita y el contacto o responsable se indica con el identificador o índice que enumera las entradas en los modelos CI-1 y CI-2 que recogen los detalles de las citas y grupos responsables respectivamente. La Figura 1 muestra los modelos por secciones del metadato.

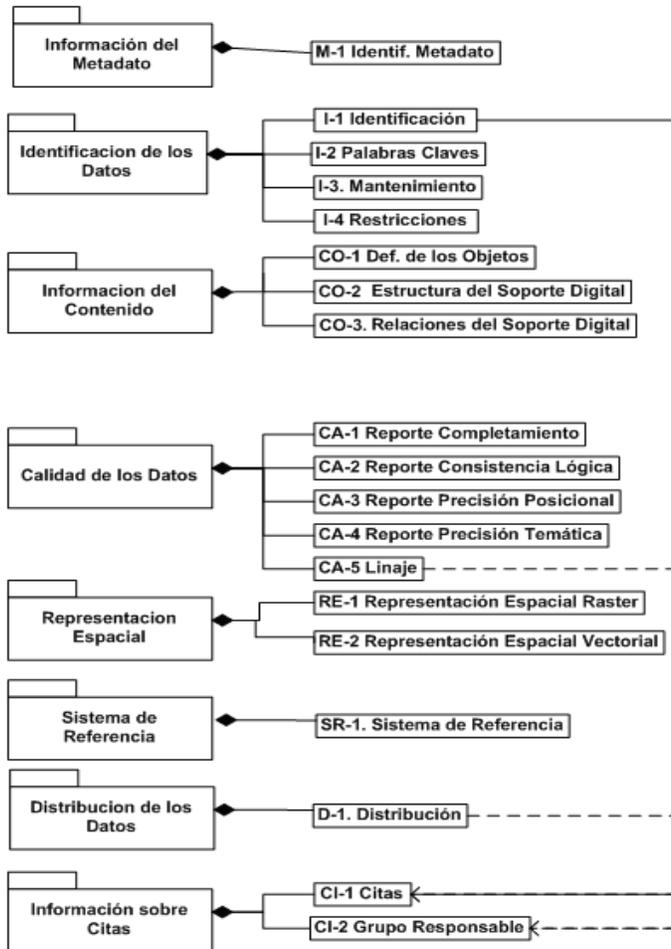


Figura 1: Metadato. Secciones y modelos de documentación.

1.3.2 Descripción actual del dominio del problema.

Como resultado de un arduo trabajo de investigación, de consultas y de intercambio de experiencias el 23 de Enero de 1995 se publicó en la Gaceta Oficial la Ley 76, “Ley de Minas”, que en su artículo 11 plantea la creación de la ONRM como autoridad minera, una institución con personalidad jurídica, adscrita al MINBAS.

Capítulo 1: Fundamentación Teórica

En el 2001 el Comité Ejecutivo del Consejo de Ministros de Cuba aprobó las funciones de la ONRM, con el objetivo de mejorar la administración del conocimiento geológico y toda la información geológica, minera y petrolera de la nación, entre sus misiones fundamentales se pueden encontrar:

- Fiscalizar y controlar la actividad minera y el uso racional de los recursos minerales, según lo dispuesto en la legislación vigente.
- Fiscalizar y controlar las actividades de exploración-producción de hidrocarburos líquidos y gaseosos, según las disposiciones legales vigentes.
- Asesorar al Ministerio de la Industria Básica y demás organismos de la administración central del estado sobre las actividades mineras de exploración producción de hidrocarburos, sin perjuicio de sus debidas competencias.
- Aprobar los cálculos de reservas minerales sólidos, líquidos y gaseosos y registrar y mantener actualizado el inventario de dichas reservas, así como certificar el grado de aprobación de dichas reservas para su asimilación industrial u otros fines.
- Ser el depositario de la información geológica, minera y petrolera de la nación, recibir, organizar y conservar la información, así como brindar servicios de información técnica.
- Proponer y en su caso dictar las disposiciones de carácter técnico que regulen la investigación, la explotación, el procesamiento y el uso de los recursos minerales.
- Proponer y en su caso dictar las disposiciones técnicas que regulen los trabajos de exploración-producción de petróleo, basadas en el óptimo conocimiento de las características geológicas y el potencial petrolífero de las zonas objeto de exploración, así como en la protección y el uso racional de las reservas de las reservas de hidrocarburos.
- Controlar la ejecución de los planes de preservación del medio ambiente y de las medidas para mitigar el impacto ambiental establecidas por los organismos competentes a las entidades que en el territorio nacional o en la zona económica exclusiva de Cuba realicen actividades mineras y petroleras.
- Mantener actualizadas las estadísticas mineras y de exploración-producción de petróleo y gas del país.

De manera general la ONRM es la entidad geológica encargada de administrar y gestionar el conocimiento geológico obtenido a lo largo de más de 40 años. Así mismo debe garantizar la creación de metadatos geológicos que cumplan con los estándares existentes a nivel mundial, para de esta forma asegurar la calidad de los datos.

Capítulo 1: Fundamentación Teórica

Actualmente existen, en el país, varias empresas geológicas, encargadas de recoger los datos que conformarán los metadatos geológicos del país, pero este proceso está actualmente retrasado y solo algunas provincias, como Pinar del Río, Camagüey y otras provincias orientales tiene definidos sus metadatos, el resto se encuentra actualmente inmersas en la creación de los metadatos geológicos.

1.3.3 Situación problemática.

Uno de los principales problemas que afectan en la actualidad el funcionamiento de la ONRM, es la recuperación de información para conformar los metadatos geológicos, pues la misma se encuentra distribuida por todo el territorio nacional y en distintos formatos lo que provoca la pérdida de información sensible, el difícil procesamiento de la información y la deficiente accesibilidad a los mismos.

La deficiente accesibilidad a los metadatos geológicos, los cuales contienen información geológica del país, provoca retrasos en los convenios de trabajo y en el establecimiento de negocios geológicos, violaciones de la ley y el desaprovechamiento de clientes potenciales. La ONRM exige la incorporación de sistemas informáticos debidamente protegidos y asegurados, robustos, que sigan estándares informáticos de calidad y normas requeridas por la geociencia, que apoyen la gestión, la comunicación, la información y sirvan de apoyo al establecimiento de mecanismos de control y seguimiento del conocimiento geológico.

La ONRM necesita garantizar la perdurabilidad de la información del conocimiento geológico del país, incrementar la velocidad de localización de los datos geológicos, mejorar el procesamiento de los datos dentro de la oficina o por sus clientes, ofrecer mejor eficiencia en cuanto a la accesibilidad de la información geológica, mantener un control adecuado sobre el acceso a la información asegurando su integridad. Esta institución requiere de una solución inmediata a los problemas antes mencionados para evitar de esta forma el deterioro de la información, logrando formatos estándares en su almacenamiento y una digitalización adecuada para conformar el Metadato a través de los datos geológicos, propiciando un nivel de seguridad óptimo para todos estos datos. Como consecuencia surge la necesidad de crear un sistema que cumpla con las normas internacionales para la creación de metadatos geológicos en la ONRM.

1.4 Otras soluciones existentes.

1.4.1 GeoNetwork.

Capítulo 1: *Fundamentación Teórica*

Herramienta “Open Source” para la creación y gestión de metadatos, desarrollada por la FAO (Organización de las Naciones Unidas para la Agricultura y la Alimentación) y el Programa de las Naciones Unidas para el Medio Ambiente conocido por sus siglas en inglés como UNEP. (18)

Es un Estándar basado en código abierto, es una aplicación libre que permite catalogar recursos referenciados espacialmente a través de la Web. Proporciona potentes funciones para la gestión y búsqueda de metadatos a partir de aplicaciones clientes así como un mapa interactivo. Estas aplicaciones están estructuradas en módulos y pueden ser personalizadas a partir de plantillas de Lenguaje de Hoja de Estilos Extensible conocido por sus siglas en inglés XSL (eXtensible Stylesheet Language). La última versión de GeoNetwork, la 4.2.2, tiene implementada la ISO19139.

Sus principales características son:

- Editor y buscador de metadatos
- Creador de un Catálogo de metadatos.
- Soporta los estándares ISO19115, Comité Federal de Datos Geográficos conocido por sus siglas en inglés FGDC, Dublín Core.
- Creación de fichero de transformación basado en hojas de estilo conocido por sus siglas en inglés XSLT.
- Multilingüe.

1.4.2 CatMDEdit

Es una herramienta “Open Source”, desarrollada por el consorcio de Tecnologías para las Infraestructuras de Datos Espaciales (TelDE) y bajo el apoyo de diferentes proyectos, de edición de metadatos que facilita la documentación de los recursos, con especial atención a la descripción de los recursos de información geográfica.

La herramienta ha sido implementada en Java y tiene las siguientes características:

- Multi-plataforma (Windows, Unix). Tal como se ha desarrollado en Java y el almacenamiento de los registros de metadatos es gestionado directamente por el sistema de ficheros, la aplicación se puede instalar en cualquier plataforma con el mínimo requisito de tener instalada una máquina virtual Java.
- Multilingüe. La aplicación se ha desarrollado siguiendo la metodología de internacionalización de Java. Hoy en día, hay versiones en español, inglés, francés, polaco, portugués y checo.

Capítulo 1: Fundamentación Teórica

- Definición y gestión de los diferentes repositorios de metadatos (depósitos también pueden contener ficheros de datos), incluyendo la selección y filtrado de los registros de metadatos almacenados en el repositorio de metadatos de cada local.
- Edición de metadatos de acuerdo con "ISO19115. Información Geográfica - Metadatos" estándar (ISO19115: 2003/Core 1 2006, ISO / TS 19139:2007 codificación XML). Edición de interfaces adaptados a diferentes perfiles de metadatos.
- Edición de metadatos, basado en el estándar Dublín Core (ISO 15836).
- Personalización de la herramienta de apoyo a los nuevos estándares de metadatos y perfiles de acuerdo a las necesidades de los usuarios.
- Generación automática de metadatos para algunos formatos de archivo de datos: Shapefile, DGN, ECW, FICC, GeoTiff, GIF / GFW, JPG / JGW, PNG / PGW. Generación automática de metadatos para la ordenación del territorio serie.
- Intercambio de registros de metadatos de acuerdo a diferentes normas en XML.

Como principales ventajas:

- CatMDEdit tiene una mayor interoperabilidad con otras Infraestructuras de Datos Espaciales o IDE que utilicen la misma herramienta (entre ellas la IDE española). Esta interoperabilidad está entendida en cuanto a que cualquier cambio en los estándares ISO 19139 e ISO19115 (todavía no definitivos) que pudiera repercutir en la herramienta, será cambiado en un corto plazo de tiempo y los mismos inconvenientes que se puedan tener en la catalogación, serán encontrados en otras comunidades.
- La herramienta CatMDEdit aporta tesauros, con un interfaz amigable para su utilización.
- Esta herramienta es muy extensa y recoge la mayoría de los metadatos de la norma ISO19115, resultando, por lo tanto, muy precisa.

Como principales inconvenientes:

- No hay una relación directa de los datos con los metadatos.
- Esta falta de relación directa entre los datos y los metadatos hace que en la versión simple de la herramienta CatMDEdit, no se pueda extraer automáticamente cierta información de los datos, como las coordenadas límite, el sistema de referencia y el sistema de coordenadas, los conjuntos de caracteres de los datos y de los metadatos, etc.
- El CatMDEdit resulta una herramienta complicada en cuanto a la diversidad de campos que hay para completar.

Capítulo 1: Fundamentación Teórica

1.4.3 MetaD

Programa de edición y exportación de metadatos siguiendo el perfil IDEC (Infraestructura de datos espaciales de Cataluña), subconjunto del estándar ISO 19115, con su implantación ISO 19139 destinado a describir la Información Geográfica (gráfica, alfanumérica, entre otras).

Es una aplicación que tiene como objetivo principal la creación de metadatos de forma guiada, de tal manera que facilite, además de la creación, la edición, el mantenimiento y la exportación de metadatos de la manera más sencilla posible para el usuario.

La estructura y terminología de este programa sigue el Estándar ISO 19115 con su implantación ISO 19139. Debido a que este estándar es muy genérico, la Infraestructura de Datos Espaciales de Cuba o IDEC, lo ha adaptado a las particularidades de Cataluña, dando como resultado los elementos que aparecen en la aplicación.

La idea es proporcionar una herramienta a los organismos o empresas interesadas en mejorar el rendimiento interno, que les permita gestionar correctamente su propia **información. Los metadatos pueden sufrir ciertos cambios, que han de poder ser explicados e incorporados. Por esta razón, el programa permite, en todo momento, recuperar cualquier registro antiguo de metadatos y realizar las modificaciones pertinentes.**

1.4.4 Comparación de las soluciones existentes.

Solución	Idioma	Interfaz Usuario	Estándar
GeoNetwork	Multilingüe	Compleja	ISO-19115
CatMDEdit	Multilingüe	Amigable	ISO-19115
MetaD	Español	Compleja	ISO-19115 (Cataluña)

Tabla 1 Comparación de las soluciones existentes.

A pesar de que todas estas herramientas permiten la creación de metadatos geológicos, es importante mencionar que ninguna de ellas tiene implementado el estándar creado en Cuba para la creación de metadatos geológicos, el cual cumple con todas las especificaciones planteadas en la norma internacional ISO 19115.

Además todas las herramientas expuestas anteriormente tienen una interfaz de usuario poco amigable para los clientes finales, lo cual hace engorroso el trabajo con las mismas y que el personal que las use deba tener una capacitación previa para poder comenzar a utilizarlas con mayores beneficios, solamente

Capítulo 1: Fundamentación Teórica

el CatMDEdit tiene una interfaz de usuario amigable, pero es muy compleja de usar pues es una herramienta muy extensa que recoge la mayoría de los metadatos que se plantean en la ISO 19115.

1.5 Conclusiones parciales.

El estudio detallado sobre los fundamentos teóricos de la creación de Metadatos Geológicos, permitió desarrollar el marco conceptual de la investigación.

Debido al monitoreo de las aplicaciones a nivel internacional y a la inexistencia de una de este tipo en Cuba, se definió que la construcción de un sistema informático capaz de crear Metadatos Geológicos para la ONRM y las empresas geológicas del país, para agilizar y lograr estándares en este proceso.

Capítulo 2: Tendencias y Tecnologías

Capítulo 2:

Tendencias y tecnologías actuales a desarrollar.

2.1 Introducción.

En el presente capítulo se verán las principales tecnologías y herramientas que se usaron en la presente investigación, mencionando características y ventajas de cada una de ellas, para un mayor dominio de las mismas y llegar a realizar aplicaciones eficientes que cumplan con las necesidades de la ONRM.

2.2 Las Tecnologías de la Información y las Comunicaciones.

Las Tecnologías de la Información y las Comunicaciones(TIC) son aquellas que se necesitan para la gestión y transformación de la información, en particular el uso de las computadoras y los software que permiten crear, modificar, almacenar, proteger y recuperar información. Las TIC constituyen un elemento esencial de la Sociedad de la Información, estas brindan la capacidad universal de acceder y contribuir la información, las ideas y el conocimiento. Promover el intercambio y el fortalecimiento de los conocimientos mundiales en favor del desarrollo permitiendo un acceso equitativo a la información para actividades económicas, sociales, políticas, sanitarias, culturales, educativas y científicas dando acceso a la información que está en el dominio público sólo son posibles gracias a las TIC. **(9)**

El uso de las TIC en las empresas posibilita una mejora de los procesos productivos internos al proporcionar herramientas que facilitan la clasificación, organización, manejo y filtro de la información, y de cara al exterior mejoran los procesos de interacción con los agentes externos: clientes, proveedores y socios, permitiendo la independencia del tiempo y del espacio, posibilitando el acceso desde cualquier lugar y a cualquier hora, abriendo la posibilidad de nuevos negocios en la red.

Principales aportes de las TIC.

Fácil acceso a todo tipo de información, sobre cualquier tema y en cualquier formato (textual, icónico, sonoro), especialmente a través de la televisión e Internet pero también mediante el acceso a las numerosas colecciones de discos en soporte CD-ROM y DVD: sobre turismo, temas legales, datos económicos, enciclopedias generales y temáticas de todo tipo, películas y vídeos digitales (se están digitalizando en soporte DVD toda la producción audiovisual), bases de datos fotográficas...

La información es la materia prima que se necesita para crear conocimientos con los que afrontar las problemáticas que se van presentando cada día en el trabajo, en el ámbito doméstico, al reflexionar.

Capítulo 2: Tendencias y Tecnologías

Instrumentos para todo tipo de proceso de datos. Los sistemas informáticos, integrados por ordenadores, periféricos y programas, permiten realizar cualquier tipo de proceso de datos de manera rápida y fiable: escritura y copia de textos, cálculos, creación de bases de datos, tratamiento de imágenes. Para ello se dispone de programas especializados: procesadores de textos, editores gráficos, hojas de cálculo, gestores de bases de datos, editores de presentaciones multimedia y de páginas web, que ayudan especialmente a expresar y desarrollar la creatividad, realizar cálculos y organizar la información

Canales de comunicación inmediata, sincrónica y asíncrona, para difundir información y contactar con cualquier persona o institución del mundo mediante la edición y difusión de información en formato web, el correo electrónico, los servicios de mensajería inmediata, los foros telemáticos, las videoconferencias, entre otros medios.

Automatización de tareas, mediante la programación de las actividades que se desea que realicen los ordenadores, que constituyen el cerebro y el corazón de todas las TIC. Ésta es una de las características esenciales de los ordenadores, que en definitiva son "máquinas que procesan automáticamente la información siguiendo las instrucciones de unos programas".

Interactividad. Los ordenadores permiten "dialogar" con programas de gestión, videojuegos, materiales formativos multimedia, sistemas expertos específicos... Esta interacción es una consecuencia de que los ordenadores sean máquinas programables y sea posible definir su comportamiento determinando las respuestas que deben dar ante las distintas acciones que realicen ante ellos los usuarios.

Homogeneización de los códigos empleados para el registro de la información mediante la digitalización de todo tipo de información: textual, sonora, icónica y audiovisual. Con el uso de los equipos adecuados se puede captar cualquier información, procesarla y finalmente convertirla a cualquier formato para almacenarla o distribuirla. Así por ejemplo, hay programas de reconocimiento de caracteres que leen y convierten en voz los textos, programas de reconocimiento de voz que escriben al dictado, escáneres y cámaras digitales que digitalizan imágenes.

2.3 Arquitectura de software.

La Arquitectura de Software se puede definir como una representación independiente de un sistema de software determinado. Representa el esqueleto de un sistema de software, haciendo hincapié en una definición clara de la estructura, la comunicación y la interrelación del cuerpo de los componentes que cumple con el propósito de un sistema de software determinado. Los componentes de la arquitectura debe

Capítulo 2: Tendencias y Tecnologías

exponer el protocolo de los componentes sólo a sus clientes. La arquitectura de software debe implementar una separación clara de las preocupaciones de todos los observables en el núcleo y el comportamiento no observables de un sistema de software. La arquitectura de la interoperabilidad de todos los comportamientos básicos para satisfacer las necesidades de los usuarios finales del sistema. La misma debe apoyar las configuraciones, tanto estáticas como dinámicas, de componentes a través de un conjunto de interfaces bien documentados o contratos conociendo a los clientes de los de un sistema de software determinado. Debe ser descrita en un conjunto de patrones para servir de claridad y como un simple medio de comunicación a las partes interesadas del sistema. La arquitectura de software debe apoyar variabilidad, la portabilidad, escalabilidad, reutilización para mencionar algunos. **(10)**

Se puede decir entonces que la arquitectura de software es una vista estructural de alto nivel, ocurre muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requerimientos no funcionales.

2.3.1 Arquitectura Modelo-Vista-Controlador

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Aunque 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, el mismo no está estrechamente vinculado con este tipo de aplicaciones, pues al ser un estilo arquitectónico, propone una vista estructural de alto nivel. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. **(11)**

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y probablemente, a la vista

2.4 Lenguaje Unificado de modelado.

El éxito de un proceso de desarrollo de software está en la comunicación que se tiene entre quien tiene la idea y quien desarrolla el software. El Lenguaje Unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje que cumple con esta función, debido a que prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas. Dado estas características

Capítulo 2: Tendencias y Tecnologías

contribuye a capturar la idea de un sistema para posteriormente comunicarla a quien esté involucrado en el proceso de desarrollo; esto se lleva a cabo mediante símbolos y diagramas. UML se ha convertido en el estándar para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Sus creadores pretendieron con este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar. **(12)**

UML es, probablemente, una de las innovaciones conceptuales en el mundo tecnológico del desarrollo de software que más expectativa ha generado a lo largo de muchos años, comparable con la aparición e implantación de los lenguajes COBOL8, Basic, Pascal, C++, y actualmente con los más recientes Java, XML, C#. UML es ya un estándar de la industria del software, pero no sólo de la industria sino, que en general, de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posteriormente para la construcción de prototipos. Además permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarla a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y Cliente/Servidor.

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Además UML es desde finales de 1997, orientado a objeto de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes empresas de desarrollo de software tales como: Microsoft, Oracle, Rational. **(13)**

2.5 Metodologías de Desarrollo de Software.

Las metodologías de desarrollo de software surgen para guiar a las personas implicadas en el desarrollo de software, brindando un conjunto de procedimientos, técnicas y herramientas, de forma que sepan qué hacer en cada momento y cómo alcanzar un producto de alta calidad. Entre las más utilizadas se puede

Capítulo 2: Tendencias y Tecnologías

encontrar el Proceso Unificado de Software (RUP), la Programación Extrema (XP) y Microsoft Solution Framework (MSF).

2.5.1 RUP

RUP (Proceso Unificado de Rational) provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. **(14)**

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto, con 3 características esenciales, está dirigido por los casos de usos: que orientan al proyecto a la importancia para los usuarios y lo que este quiere; está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden; y es iterativo e incremental: dividiéndose el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen su uso de manera más depurada.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en un número variable según el proyecto y en las que se hace un mayor o menor hincapiés en las distintas actividades

Ventajas:

- Evaluación en cada fase que permite cambios de objetivos.
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

2.5.2 XP

XP (Programación Extrema), es la metodología más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. **(14)**

Las características fundamentales de la programación extrema son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.

Capítulo 2: Tendencias y Tecnologías

- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad y hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo. **(14)**

Ventajas:

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para nuestros clientes, conocen las fechas de entrega de funcionalidades.
- Permitirá definir en cada iteración cuales son los objetivos de la siguiente.
- Permite tener realimentación de los usuarios muy útil.
- La presión está a lo largo de todo el proyecto y no en una entrega final. **(14)**

2.5.3 MSF.

MSF (MICROSOFT SOLUTION FRAMEWORK), es un modelo para el ciclo de vida de un software creado por la compañía Microsoft en el cual se nota un uso de framework. El framework es una estructura o un marco de trabajo utilizado para el desarrollo del software.

Capítulo 2: Tendencias y Tecnologías

Este modelo además posee una flexibilidad que hace posible que sea usado tanto en proyectos grandes como en proyectos pequeños (refiriéndose al alcance de los proyectos). **(15)**

El MSF está compuesto por las siguientes fases:

- Visión.
- Planeación.
- Desarrollo.
- Estabilización.
- Instalación.
- Soporte.

Este modelo al ser creado por la Microsoft tiene términos tales como estabilización refiriéndose a la fase de pruebas; e instalación que personalmente es el primer modelo que usa esta palabra.

Una desventaja muy importante de MSF es que este modelo requiere una gran cantidad de documentación, pero esta desventaja es relativa ya que tiene dos aspectos:

- El primero es que la documentación puede no ser lo suficientemente específica y provocar al usuario del MSF una cierta confusión.
- El segundo aspecto y el cual contrasta con el anterior es el hecho de que en el momento de realizar la documentación, el desarrollador poseerá un conocimiento más amplio sobre el software en desarrollo.

(15)

2.5.4 Selección de la metodología a utilizar.

Luego del análisis de las metodologías existentes a nivel mundial se selecciona RUP para la realización del trabajo, por ser la que más se adapta a las características del proyecto. Se puede decir que no existen dos proyectos de desarrollo de software que sean iguales. Cada uno tiene prioridades, requerimientos, y tecnologías muy diferentes. Sin embargo, en todos los proyectos, se debe minimizar el riesgo, garantizar la predictibilidad de los resultados y entregar software de calidad superior a tiempo. Rational Unified Process, o RUP, es una plataforma flexible de procesos de desarrollo de software que ayuda brindando guías consistentes y personalizadas de procesos para todo el equipo de proyecto.

RUP describe cómo utilizar de forma efectiva reglas de negocio y procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como “mejores prácticas”. Captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones. Es una guía de cómo utilizar de manera

Capítulo 2: Tendencias y Tecnologías

efectiva UML. Provee a cada miembro del equipo fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo. Crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación.

2.6 Lenguajes de programación

Los lenguajes de programación son herramientas que permiten crear programas y software. Una computadora funciona bajo el control de un programa el cual debe estar almacenado en la unidad de memoria; tales como el disco duro. Los lenguajes de programación facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar. **(16)**

2.6.1 Java.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Este es un lenguaje que ha sido diseñado para producir software, confiable, multiplataforma, la compatibilidad es total (a nivel de fuentes, a nivel de bibliotecas, a nivel del código compilado), flexible y robusto. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros. **(17)**

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque esta ejecución de manera directa por hardware del bytecode por un procesador Java también es posible.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

Características del lenguaje Java (18)

Lenguaje simple.

Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir applets interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce aunque sea de forma elemental, resulta muy fácil

Capítulo 2: Tendencias y Tecnologías

aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.

Orientado a objetos.

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

Distribuido.

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

Interpretado y compilado a la vez.

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador.

Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).

Robusto.

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

Seguro.

Dada la naturaleza distribuida de Java, donde las applets se bajan desde cualquier punto de la Red, la seguridad se impuso como una necesidad de vital importancia. A nadie le gustaría ejecutar en su ordenador programas con acceso total a su sistema, procedentes de fuentes desconocidas. Así que se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

Indiferente a la arquitectura.

Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con

Capítulo 2: Tendencias y Tecnologías

sistemas operativos diversos. Para acomodar requisitos de ejecución tan variados, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñada para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.

Portable.

La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

2.6.2 C#

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de bases de datos, y muchas tareas más. Microsoft Visual C# 2005 proporciona un editor de código avanzado, diseñadores de interfaz de usuario prácticos, un depurador integrado y otras herramientas para facilitar un rápido desarrollo de la aplicación basado en la versión 2.0 del lenguaje C# y en .NET Framework.

La sintaxis de C# es muy expresiva, aunque cuenta con menos de 90 palabras claves; también es sencilla y fácil de aprender. Está basada en signos de llave, podrá ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java. Los desarrolladores que conocen cualquiera de estos lenguajes pueden empezar a trabajar de forma productiva en C# en un plazo muy breve. La sintaxis de C# simplifica muchas de las complejidades de C++ y a la vez, ofrece funciones eficaces tales como: tipos de valores que aceptan valores NULL, enumeraciones, delegados, métodos anónimos y acceso directo a memoria, que no se encuentran en Java. C# también admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados, que el código de cliente puede utilizar fácilmente.

(19)

C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen:

- Firmas de métodos encapsulados denominadas delegados, que permiten notificaciones de eventos con seguridad de tipos.

Capítulo 2: Tendencias y Tecnologías

- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.

El proceso de generación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos. **(19)**

2.6.3 C++

El lenguaje de programación C++ fue creado en los años 80 por Bjarne Stroustrup basado en el lenguaje C. El C++ es un lenguaje orientado a objetos al que se le añadieron características y cualidades de las que carecía el lenguaje C.

C++ depende mucho del hardware, es uno de los lenguajes más potentes porque permite programar a alto y a bajo nivel, es complicado porque el programador debe hacerlo casi todo.

Este lenguaje es un lenguaje de programación mixto y se le puede compilar. Una de las ventajas que ofrece es que es mucho más sencillo de aprender para los programadores que ya conocen el lenguaje C.

Tiene una enorme compatibilidad con el C principalmente por dos razones: Por la gran cantidad de código C que comparten, y para facilitar el paso de los programadores de C al nuevo lenguaje C++. Este lenguaje no es un lenguaje orientado a objetos puro, porque nace como una evolución de otro anterior. **(20)**

Desde luego, C++ es un lenguaje de programación extremadamente largo y complejo. También se puede decir que prácticamente no hay una regla sin su correspondiente excepción. Cuando se aprende que algo no se puede hacer, hay siempre algún truco escondido para hacerlo.

A pesar de todo, ha experimentado un extraordinario éxito desde su creación. De hecho, muchos sistemas operativos, compiladores e intérpretes han sido escritos en C++ (el propio Windows y Java). Una de las razones de su éxito es ser un lenguaje de propósito general que se adapta a múltiples situaciones. **(21)**

Para mejorar el rendimiento en C++ se deben tomar una serie de precauciones por ejemplo:

- Usar enteros (int) con preferencia sobre cualquier otro tipo de variable numérica. En especial en los contadores de bucles. Las operaciones con enteros son del orden de 10 a 20 veces más rápidas que las de números en coma flotante.

Capítulo 2: Tendencias y Tecnologías

- Utilizar operadores de incremento y decremento ++/--.
- Emplear variables de registro, en especial en los bucles críticos, sobre todo si son anidados.
- Usar aritmética de punteros frente a subíndices de matrices.
- En problemas de computación numérica el cálculo de funciones trascendentes es por lo general muy lento.
- Usar referencias para argumentos y valores devueltos en funciones, antes que objetos "por valor".
- Prestar atención al modo de uso de aquellas funciones de librería que se presentan en dos versiones.

(21)

2.6.4 Selección del lenguaje de programación a utilizar

El lenguaje de programación seleccionado es Java pues es el lenguaje más utilizado a nivel mundial. Este cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Además posee código abierto, facilidad en la programación y es muy extendido. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se hace un programa en Java podrá funcionar en cualquier ordenador del mercado.

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi todo lo que se pueda hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. Con Java se puede programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desee realizar con acceso a través de web se puede hacer utilizando Java.

2.7 XML.

El lenguaje de marcas extensibles (XML, por sus siglas en inglés, Extensible Markup Language), es un lenguaje de marcado de carácter general. Creado en 1998 por W3C, es un estándar internacional libre y gratuito.

XML es extensible (*extensible*) pues ofrece al usuario la posibilidad de definir sus propios elementos: el usuario puede definir marcas o etiquetas de su gusto, y estructurar el documento en función de dichas etiquetas. De hecho, se debería decir que XML más que un lenguaje es un metalenguaje, ya que ofrece la oportunidad de definir y utilizar diferentes lenguajes o aplicaciones. XML, per se, no define más que la sintaxis, regulando cómo escribir y organizar texto y marcas en el documento.

Capítulo 2: Tendencias y Tecnologías

El objetivo fundamental de XML es intercambiar datos estructurados entre sistemas de información, fundamentalmente a través de Internet. Se trata de un formato de texto plano, lo que facilita enormemente la transferencia de información, logrando independencia con respecto a las diferentes plataformas. Utilizando XML se puede marcar cualquier documento. Además, como acepta Unicode, es válido para cualquier idioma. El estándar *Unicode* permite codificar electrónicamente caracteres escritos en cualquier lengua. XML es un formato autodocumentado que, además de por máquinas, puede ser leído e interpretado por personas.

Ventajas de XML:

Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.

El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de aplicaciones.

Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

Comunicación de datos. Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.

Migración de datos. Si se necesita mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en formato XML.

Aplicaciones web. Hasta ahora cada navegador interpreta la información a su manera y los programadores del servidor web deben hacer unas cosas u otras en función del navegador del usuario. Con XML se tiene una sola aplicación que maneja los datos y para cada navegador o soporte se obtendrá una hoja de estilo o similar para aplicarle el estilo adecuado. Si mañana nuestra aplicación debe correr en WAP solo se debe crear una nueva hoja de estilo o similar.

2.7.1 Justificación del lenguaje XML.

Además de las ventajas que brinda este lenguaje, al poder definir los elementos propios de la investigación, se decide utilizarlo, pues, la especificación técnica ISO-19139, propone una implementación del estándar ISO-19115 para la representación gráfica de los metadatos geológicos lo que provee al

Capítulo 2: Tendencias y Tecnologías

sistema a desarrollar de una tecnología novedosa y portable con la cual se podrán lograr los estándares de formatos para las empresas geológicas que generen metadatos geológicos en el país.

2.8 Herramientas CASE.

Las herramientas de Ingeniería de Software Asistida por Ordenador (CASE, por sus siglas en Inglés, Computer Aided Software Engineering) 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. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. **(22)**

Sin lugar a dudas las herramientas CASE han venido a revolucionar la forma de automatizar los aspectos clave en el desarrollo de los sistemas de información, debido a la gran plataforma de seguridad que ofrecen a los sistemas que las usan y es que éstas, brindan toda una gama de componentes que incluyen todas o la mayoría de los requisitos necesarios para el desarrollo de los sistemas, han sido creadas con una gran exactitud en torno a las necesidades de los desarrolladores de sistemas para la automatización de procesos incluyendo el análisis, diseño e implantación. **(23)**

2.8.1 Visual Paradigm.

Visual Paradigm es una herramienta CASE, para el trabajo con UML, profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar ingeniería tanto directa como inversa. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, y permite control de versiones. Visual Paradigm ofrece: **(24)**

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

Capítulo 2: Tendencias y Tecnologías

Para maximizar la interoperabilidad de los productos de Visual Paradigm con otras aplicaciones, se han introducido opciones de exportación e importación de sus modelos a otras aplicaciones, lo que posibilita que los usuarios y proveedores de tecnología puedan integrar modelos de Visual Paradigm en sus soluciones con un mínimo esfuerzo.

2.8.2 Rational Rose

Rational Rose es una herramienta para modelado visual, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Rational permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto: **(25)**

- Modelado del negocio.
- Captura de requisitos (parcial).
- Análisis y diseño (completo).
- Implementación (como ayuda).
- Control de cambios y gestión de configuración (parte).

Características principales de Rational Rose: (25)

- Admite como notaciones: UML, COM, OMT y Booch.
- Realiza chequeo semántico de los modelos.
- Ingeniería "de ida y vuelta": Rose permite generar código a partir de modelos y viceversa.
- Desarrollo multiusuario.
- Integración con modelado de dato.
- Generación de documentación.
- Tiene un lenguaje de script para poder ampliar su funcionalidad.
- Soporta OLE.
- Disponible en múltiples plataformas.

Versiones de Rational Rose: (25)

- Modeler: no tiene soporte para lenguajes de programación.
- Professional: incluye soporte sólo para un lenguaje.
- Enterprise: múltiples lenguajes, incluyendo C++, Visual C++ (6.0), Visual Basic, Java, CORBA, etc.).

2.8.3 Selección de la Herramienta CASE a utilizar.

Capítulo 2: Tendencias y Tecnologías

La herramienta CASE a utilizar en la investigación es Visual Paradigm, debido a que esta herramienta emplea las últimas notaciones del Lenguaje Unificado de Modelado, ingeniería inversa, generación de código, importación de Rational Rose, exportación/importación XML, generador de impresos e integración con el Visio. Además de lo anterior, soporta aplicaciones web, genera código para el lenguaje Java, y exporta en formato HTML, es una tecnología libre y está disponible en varios idiomas, en conjunto con esto es fácil de instalar y fácil de actualizar. Visual Paradigm admite compatibilidad con las demás versiones, es multiplataforma; además facilita la organización de los diagramas, generando la documentación del proyecto automáticamente en varios formatos como Web o PDF y permite el control de versiones.

2.9 Entornos de desarrollo.

Un entorno de desarrollo integrado (IDE por sus siglas en inglés, Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. **(26)**

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

2.9.1 NetBeans.

NetBeans IDE es un entorno de desarrollo visual para aplicaciones programadas a partir del uso del lenguaje de programación Java, de modo que puede ejecutarse en cualquier ambiente que ejecute Java, es uno de los lenguajes de programación más poderos del momento.

Es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. **(27)**

Aparte de la filosofía de distribución y desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma. En una era

Capítulo 2: Tendencias y Tecnologías

en la cual la arquitectura orientada al servicio (SOA) requiere servicios con cierta relación que manejen procesos específicos del negocio, NetBeans satisface los requisitos con un conjunto de herramientas independientes de la plataforma, modulares y orientadas al objeto.

La automatización de los requerimientos de diseño demuestra ser particularmente importante en el diseño de aplicaciones para de Arquitectura Orientada a Servicios (SOA) donde los desarrolladores trabajan generalmente con múltiples tecnologías y protocolos. Este IDE de Java es de mucha confianza para los desarrolladores y sumamente útil por su manual de instrucciones.

2.9.2 Eclipse

Eclipse es un entorno de desarrollo integrado multiplataforma (utilizado para los lenguajes C, C++, Python y Java entre otros) de código abierto, utilizado en su mayoría para desarrollar otros entornos de desarrollo (como el JDT) aunque también puede ser utilizado para desarrollar aplicaciones cliente como Azureus (un cliente de BitTorrent). Fue originalmente desarrollado por IBM para pasar a la familia de herramientas VisualAge que la marca poseía. Sin embargo en la actualidad esta herramienta está siendo desarrollada por la Fundación Eclipse, una organización independiente sin ánimo de lucro que intenta fomentar una comunidad de código abierto así como un conjunto de productos complementarios, capacidades y servicios.

La versión existente en la actualidad de esta herramienta ofrece las siguientes características: editor de texto, resaltado de sintaxis, compilación en tiempo real, pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant y asistentes para el inicio en algunos de los elementos soportados (como proyectos, clases y test). Además, a través del uso de plug-ins se pueden utilizar tanto Subversion para realizar el control de versiones como Hibernate para desarrollar las funciones de un motor de persistencia.

(26)

2.9.3 Selección del entorno de desarrollo a utilizar.

El entorno de desarrollo que se utilizará para la realización del sistema es NetBeans, la selección parte a partir de que el mismo contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente. NetBeans es un producto libre y gratuito sin restricciones de uso. Además es un multilenguaje completo y modular; igualmente es multiplataforma y permite construir aplicaciones completas para el cliente, permite crear ventanas, menús, barras de herramientas y acciones fácilmente. También se puede decir que NetBeans

Capítulo 2: Tendencias y Tecnologías

completa el código mas inteligentemente y resaltado; al igual que brinda mayores facilidades para el diseño de aplicaciones de escritorio.

2.10 Framework.

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

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

2.10.1 Framework a utilizar: Swing.

Swing es un framework de interfaz gráfica para Java. Incluye componentes para interfaz gráfica de usuario tales como cajas de texto, botones, desplegados y tablas. Swing introdujo un mecanismo que permitía que el aspecto de cada componente de una aplicación pudiese cambiar sin introducir cambios sustanciales en el código de la aplicación. La introducción de soporte ensamblable para el aspecto permitió a Swing emular la apariencia de los componentes nativos manteniendo las ventajas de la independencia de la plataforma. También contiene un conjunto de herramientas que permiten crear un interfaz atractivo para los usuarios.

Sigue un simple modelo de programación por hilos, y posee las siguientes características principales:

Independencia de plataforma.

Propone una arquitectura altamente particionada: los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto.

Configurable: dado el modelo de representación programático del framework de swing, el control permite representar diferentes aspectos visuales.

Ventajas

Capítulo 2: Tendencias y Tecnologías

El diseño en Java puro posee menos limitaciones de plataforma.

El desarrollo de componentes Swing es más activo.

Los componentes de Swing soportan más características.

2.11 Sistema Gestor de Base de Datos.

Los Sistemas Gestores de Base de Datos (SGBD) son software muy específico, encargados de servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Estos sistemas están compuestos por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consultas.

Los sistemas gestores de base de datos tienen como finalidad, manejar de una forma clara, sencilla y ordenada un grupo de datos que consecutivamente se convertirán en información, para una buena manipulación de datos. Entre los sistemas gestores de base de datos más conocidos por los desarrolladores de sistemas se encuentran el PostgreSQL, MySQL, Microsoft SQL Server, Oracle, entre otros.

Todos estos sistemas deben cumplir con una variedad de objetivos tales como: abstracción de información, independencia, redundancia mínima, consistencia, seguridad, integridad, respaldo y recuperación, control de concurrencia y tiempo de respuesta.

2.11.1 Sistema gestor de base de datos a utilizar: HSQLDB.

HSQLDB es un gestor de base de datos hecho totalmente en Java, pero que tiene características muy interesantes. Por un lado, se puede arrancar un servidor de base de datos HSQLDB y conectarse a él desde Java como a cualquier otra base de datos, utilizando, por supuesto, el driver suministrado por HSQLDB.

Por otro lado, sin necesidad de arrancar un servidor y sobre la marcha, una aplicación *java* puede conectarse a una base de datos *HSQLDB* ficticia que se guardará en fichero. De esta forma, sin servidor de base de datos, se pueden crear tablas y llenarlas de datos que se guardarán en un fichero, y luego recuperarlos. Eso sí, lo hará siempre usando el lenguaje *SQL* típico de las bases de datos, la clase *Connection*, *Statement* y *ResultSet*. No se deberán usar las clases y métodos típicos de la escritura en fichero, como *File*, *PrintWriter*, *FileOutputStream*, etc.

Capítulo 2: Tendencias y Tecnologías

Finalmente, se puede hacer lo mismo que en el caso anterior, pero sobre memoria. No se escribe ningún dato en ningún sitio y cuando la aplicación termine se perderán los datos.

2.12 Conclusiones parciales.

En el capítulo que culmina se pudo apreciar, las ventajas que tiene para la sociedad moderna el uso de las TIC, así como otras tecnologías que guiaron durante el desarrollo de la presente investigación.

Gracias a las herramientas seleccionadas se realizaron los modelos necesarios para desarrollar la aplicación deseada, así como implementarla con patrones arquitectónicos de gran impacto en los últimos tiempos.

Capítulo 3: Solución propuesta

Capítulo 3:

Presentación de la solución propuesta.

3.1 Introducción.

Para el desarrollo exitoso de la aplicación a desarrollar es necesario tener un amplio conocimiento de los procesos que ocurren en la actualidad ONRM acerca de la creación de Metadatos Geológicos, así como identificar los principales requerimientos funcionales y no funcionales que debe tener el sistema para posteriormente detallar sus principales procesos. Aspectos como los antes mencionados se comprenden mejor modelándolos mediante el conjunto de normativas y artefactos que proporciona el Proceso Unificado de desarrollo del Software; por todo esto en el presente capítulo se realizará el modelo del negocio, se identificarán y establecerán el conjunto de requisitos tanto funcionales como no funcionales existentes y se modelará el sistema completamente.

3.2 Modelo de Negocio.

Un Modelo de Negocio describe los procesos de negocio que se desarrollan dentro de un empresa en términos de Casos de Uso del Negocio y actores del negocio que se corresponden con los procesos de negocio y sus clientes respectivamente, además, presenta el entorno de la organización desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. **(28)**

Los objetivos del Modelamiento del negocio son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

3.2.1 Actores y trabajadores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Representan un conjunto coherente de roles que los usuarios de casos de uso desempeñan cuando interactúan con estos casos de uso. **(28)**. La siguiente tabla muestra los actores identificados.

Capítulo 3: Solución propuesta

Actor	Descripción
Cliente	El cliente no es más que una institución interesada en obtener determinada información geológica.
Técnico	El técnico es el encargado de iniciar las acciones de revisar el metadato y archivar metadato, al mismo tiempo es el principal beneficiado con el resultado de los procesos de negocio. Se utiliza además para representar a la persona encargada de entregar el archivo metadato al grupo de revisores y luego de ser revisado al especialista para actualizar la BD de metadatos.

Tabla 2: Descripción de los actores del negocio.

Un trabajador del negocio al igual que un actor del negocio es un abstracción de una persona (o grupo de personas), una máquina o sistemas automatizados, con la particularidad que estos actúan en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol y son posibles actores del sistema a construir. La siguiente tabla muestra los trabajadores del negocio.

Trabajador	Descripción
Especialista	Es el encargado de archivar el metadato una vez que ha sido aprobado. No se beneficia en ningún momento de las acciones realizadas en el proceso de negocio, sino que se limita a ejecutar dichas acciones.
Grupo de Revisores	Es la entidad encargada de revisar y aprobar los archivos antes de llevarlos al formato digital, basados en estándares y/o normas de calidad. No se beneficia en ningún momento de las acciones realizadas en el proceso de negocio, sino que se limita a ejecutar dichas acciones.

Tabla 3: Descripción de los trabajadores del negocio.

Capítulo 3: Solución propuesta

3.2.2 Procesos de negocio.

En el Modelamiento del Negocio se identifican tres procesos, los mismos constituyen los casos de usos del negocio que se identifican y describen, a continuación se muestran estos casos de usos con sus descripciones:

Caso de Uso del Negocio Consultar Metadato: cuando un cliente llega a la ONRM y solicita consultar determinada información archivadas en metadatos, el especialista de la oficina realiza las verificaciones pertinentes y si el cliente puede acceder a la información solicitada se la muestra, para que el mismo pueda consultarla.

Caso de Uso del Negocio Archivar Metadato: el técnico solicita archivar digitalmente el archivo metadato, el especialista verifica su certificado de calidad y procede a archivar el mismo digitalmente.

Caso de Uso del Negocio Revisar Metadato: cuando el técnico presenta el archivo metadato para ser revisado, el Grupo de Revisores da revisión a dicho archivo velando por el cumplimiento de las normas ISO para el metadato y entrega un certificado que avala la calidad del mismo, terminado así el caso de uso.

3.2.3 Diagrama de Casos de Uso del Negocio.

La siguiente figura muestra el diagrama de casos de usos del negocio, conformado por los actores del negocio y los principales procesos identificados anteriormente y las relaciones entre los mismos.

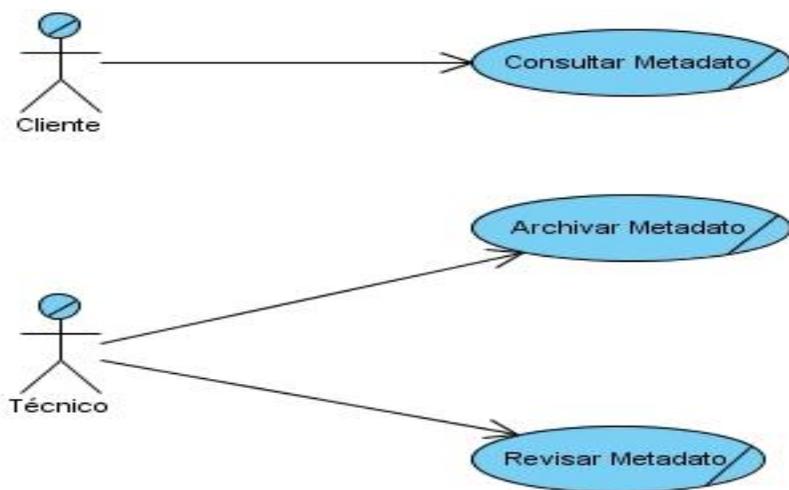


Figura 2: Diagrama de Casos de Usos del Negocio.

Capítulo 3: Solución propuesta

3.2.4 Descripción textual de los Casos de Uso del Negocio.

3.2.4.1 Descripción Textual del Caso de Uso: Consultar Metadato.

Caso de uso del negocio	Consultar Metadato
Actores	Cliente
Resumen	El caso de uso inicia cuando un cliente llega a la Oficina Nacional de Recursos Minerales y solicita consultar determinada información archivadas en metadatos, el especialista de la oficina realiza las verificaciones pertinentes y en dependencia de estas otorga o no la información solicitada.
Flujo Normal de Eventos	
Acción del actor	Respuesta del proceso de negocio
1. El caso de uso inicia cuando el cliente llega a la ONRM y solicita consultar determinada información archivada en metadatos. 2. Entrega los documentos de identificación. 3. Recibe la información y se retira.	1.1. El especialista solicita los documentos de identificación. 2.1. Verifica los documentos de identificación. 2.2. Verifica el tipo de información solicitada. 2.3. Verifica si el cliente está autorizado a recibir la información solicitada. 2.4. El especialista procede a localizar la información. 2.5. Entrega la información solicitada.
Flujos Alternos	
Acción del actor	Respuesta del proceso de negocio
	2.3. Si no está autorizado el cliente para consultar la información solicitada, se le

Capítulo 3: Solución propuesta

	notifica.
3. El cliente se retira.	
Pos condiciones	Se consulta un metadato.

Tabla 4: Descripción Textual del Caso de Uso: Consultar Metadato.

3.2.4.2 Descripción Textual del Caso de Uso: Archivar Metadato.

Caso de uso del negocio	Archivar Metadato
Actores	Técnico.
Resumen	El caso de uso comienza cuando el técnico solicita archivar digitalmente el archivo metadato, el especialista verifica su certificado de calidad y procede a ejecutar la acción de archivar, terminando así el caso de uso.
Flujo Normal de Eventos	
Acción del actor	Respuesta del proceso de negocio
1. El técnico solicita al especialista archivar metadato.	1.1. El especialista recoge el archivo metadato y su certificación. 1.2. El especialista verifica la validez del certificado. 1.3. Si el certificado es fidedigno entonces se archiva digitalmente. 1.4. Después de almacenado el especialista informa del éxito del proceso.
Flujos Alternos	
Acción del actor	Respuesta del proceso de negocio
	1.3. Si el certificado no es fidedigno se le entrega al técnico el archivo para su revisión con el Grupo de Revisores.
2. El técnico recibe el archivo y se marcha.	
Pos condiciones	Queda archivado el metadato.

Tabla 5: Descripción Textual del Caso de Uso: Archivar Metadato.

Capítulo 3: Solución propuesta

1.1.1.1 Descripción Textual del Caso de Uso: Revisar Metadato.

Caso de uso del negocio	Revisar Metadato.
Actores	Técnico.
Resumen	El caso de uso comienza cuando el técnico presenta el archivo metadato para ser revisado, el Grupo de Revisores da revisión a dicho archivo velando por el cumplimiento de las normas ISO para el metadato y entrega un certificado que avala la calidad del mismo, terminado así el caso de uso.
Flujo Normal de Eventos	
Acción del actor	Respuesta del proceso de negocio
1. El técnico solicita al Grupo de Revisores la revisión del archivo metadato.	1.1. El Grupo de Revisores solicita la entrega del archivo de metadato.
2. El técnico hace entrega del archivo creado.	2.1. Grupo de Revisores hace la revisión del archivo metadato haciendo uso de los estándares de documentación del metadato. 2.2. Si el archivo está correcto se crea un certificado que avale su calidad y se entrega.
3. El técnico recibe el archivo revisado y su certificación.	
Flujos Alternos	
Acción del actor	Respuesta del proceso de negocio
	2.2. Si el archivo no está correcto se deniega y se le entrega al técnico para su correcta elaboración.
3. El técnico recibe el archivo y se marcha.	

Capítulo 3: Solución propuesta

Pos condiciones	Queda revisado el archivo metadato y se crea su respectivo certificado de calidad.
-----------------	--

Tabla 6: Descripción Textual del Caso de Uso: Revisar Metadato.

3.3 Requisitos funcionales.

Los requisitos funcionales son las acciones que el sistema debe ser capaz de realizar, especifican el comportamiento de entrada y salida del sistema. Los requisitos funcionales no son más que:

- Las especificaciones de la funcionalidad del sistema.
- Acciones que el producto debe realizar: chequear, calcular, almacenar, recuperar, etc.
- Derivados del objetivo fundamental del producto
- Los requisitos funcionales de la presente propuesta son:

R 1. Gestionar datos del metadato.

R 1.1. Adicionar datos del metadato.

R 1.2. Modificar datos del metadato.

R 1.3. Eliminar datos del metadato.

R 2. Gestionar datos de responsables.

R 2.1. Adicionar datos del responsable.

R 2.2. Modificar datos del responsable.

R 2.3. Eliminar datos del responsable.

R 3. Generar XML.

R 4. Exportar XML.

R 5. Cargar Configuración.

3.4 Requisitos no funcionales.

Un requisito no funcional especifica propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de las plataformas, extensibilidad o fiabilidad. Un requisito puede especificar restricciones físicas sobre un requisito funcional. **(28)**. A continuación se listan los requisitos no funcionales del sistema propuesto.

Requisitos de usabilidad.

- El sistema debe poder ser usado por cualquier persona que tenga conocimientos básicos de computación.

Capítulo 3: Solución propuesta

- Debe poseer una interfaz amigable y que muestre de forma clara todas las funcionalidades que brinda la aplicación.

Requisitos de apariencia o interfaz externa.

- El sistema debe tener una apariencia profesional y un diseño gráfico sencillo, composición homogénea de color, estilos y fuentes en toda la interfaz.

Requisitos de portabilidad y operatividad.

- La aplicación debe ser compatible con los Sistemas Operativos: Windows 2000 NT, Windows XP y GNU/Linux.

Requisitos de software.

- Las computadoras que utilizarán el software deben tener instalado:
- Windows 2000 NT, Windows XP Profesional ó GNU/Linux en cualquier distribución.
- El JRE para poder ejecutar la aplicación, en su versión 1.6 o superior.

Requisitos de hardware.

- Las computadoras que utilizarán el software deberán tener 256 MB de memoria RAM como mínimo.

3.5 Descripción del sistema propuesto.

3.1.1 Descripción de los actores del sistema.

Los actores del sistema pueden ser terceros fuera del sistema que interactúan con él. Además cada trabajador del negocio (incluso si fuera un sistema ya existente) que tiene actividades a automatizar, es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

Los actores del sistema:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

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

Capítulo 3: Solución propuesta

Especialista	El especialista, ya sea un geólogo o un operador es el encargado de gestionar los metadatos y realizar cualquier otra operación de búsqueda sobre los metadatos.
--------------	--

Tabla 7: Descripción de los actores del sistema.

3.1.2 Diagrama de casos de usos del sistema.

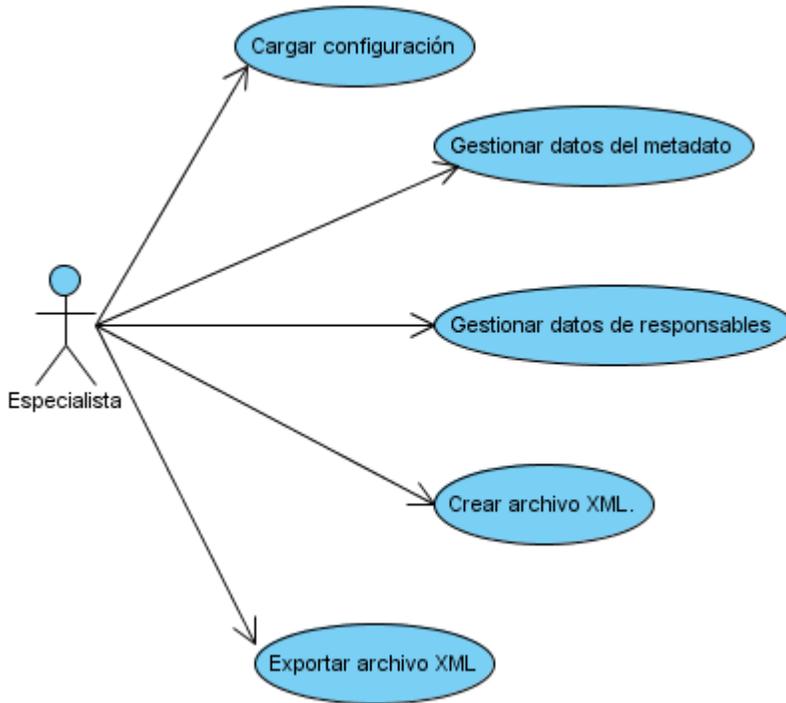


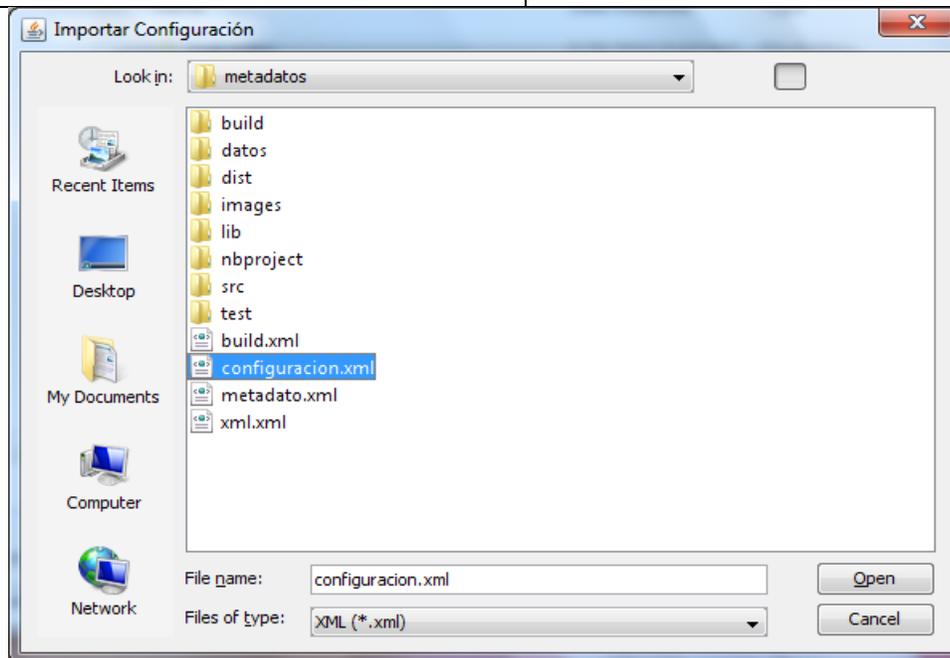
Figura 3: Diagrama de casos de usos del sistema.

3.1.3 Descripción de los casos de usos del sistema.

Caso de Uso:	Cargar configuración.
Actores:	Especialista.
Resumen:	El CU se inicia cuando el especialista desea cargar en la aplicación la configuración de la misma, la cual contiene los datos de los nomencladores actualizados y las referencias a los datos que necesita la aplicación.
Precondiciones:	Debe estar creado el archivo de configuración y estar disponible localmente.
Referencias	RF 5.
Prioridad	Crítico.

Capítulo 3: Solución propuesta

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El especialista selecciona la opción “Cargar Configuración”.	1.1 El sistema muestra una pantalla para que el especialista seleccione la ubicación del fichero de configuración.
2. El especialista selecciona la ubicación y el nombre del fichero.	2.1 El sistema comprueba que la ubicación sea correcta. 2.2 El sistema comprueba que el nombre sea correcto. 2.3 El sistema comprueba que el archivo sea un archivo de configuración válido. 2.4 El sistema Muestra al usuario el mensaje “Configuración cargada exitosamente”.



Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	2.1 Si la ubicación no es correcta el sistema muestra el mensaje de error: ""
4. El especialista presiona el botón Cancelar.	4.1 El sistema vuelve al paso 2.
Pos condiciones	Queda cargada la configuración de los nomencladores en la aplicación.

Capítulo 3: Solución propuesta

Tabla 8 Descripción textual del CU cargar configuración.

Tabla 9 Descripción textual de CU Gestionar datos de responsables.

El resto de las descripciones textuales se muestran en el Anexo 3.

3.6 Conclusiones parciales.

En el presente capítulo se modeló completamente lo referente a los diferentes procesos del negocio de la creación de Metadatos Geológicos en la ONRM, se identificaron los requisitos tanto funcionales como no funcionales existentes y posteriormente se modeló el sistema a desarrollar basado en sus casos de usos. Todo lo expuesto anteriormente conlleva a que el sistema a desarrollar pueda transitar con eficiencia en los siguientes flujos de trabajo que proporciona el Proceso Unificado de Desarrollo.

Capítulo 4: Solución propuesta

Capítulo 4:

Modelación del Diseño e Implementación de la solución propuesta.

4.1 Introducción.

Realizar un diseño para una aplicación es muy importante, pues, con ello se puede desarrollar la programación del sistema de forma más eficiente, pues permite un acercamiento al lenguaje de programación en el cual será desarrollada. En el presente capítulo se modela el diseño e implementación del sistema propuesto, con todos sus artefactos y diagramas basándose en distintos patrones de diseño y arquitectura.

4.2 Modelo de diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como entrada fundamental de las actividades de implementación.

Durante el modelo de diseño se debe tener en cuenta la arquitectura propuesta (MVC), para poder aplicar patrones de diseño correctamente.

4.2.1 Patrones utilizados durante el diseño.

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y ofrece una solución a ese problema, en tal sentido es posible usar esa solución millones de veces, sin tener que hacer lo mismo dos veces. **(29)**

Patrón “Data-Mapper”: este patrón es una capa que separa los objetos que se cargan en memoria de la base de datos, garantizando total independencia del sistema gestor que se utilice. **(30)** Este patrón propone que se creen dos clases por cada tabla de la base de datos una que contiene los métodos de acceso a datos y otra que solamente tenga los atributos de la tabla con sus métodos de acceso.

Instancia única (Singleton): El patrón Singleton garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ésta instancia. Ventajas:

- Acceso controlado a la única instancia. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.

Capítulo 4: Solución propuesta

- Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton.

4.2.2 Diagrama de clases del diseño del CU Gestionar datos de responsables.

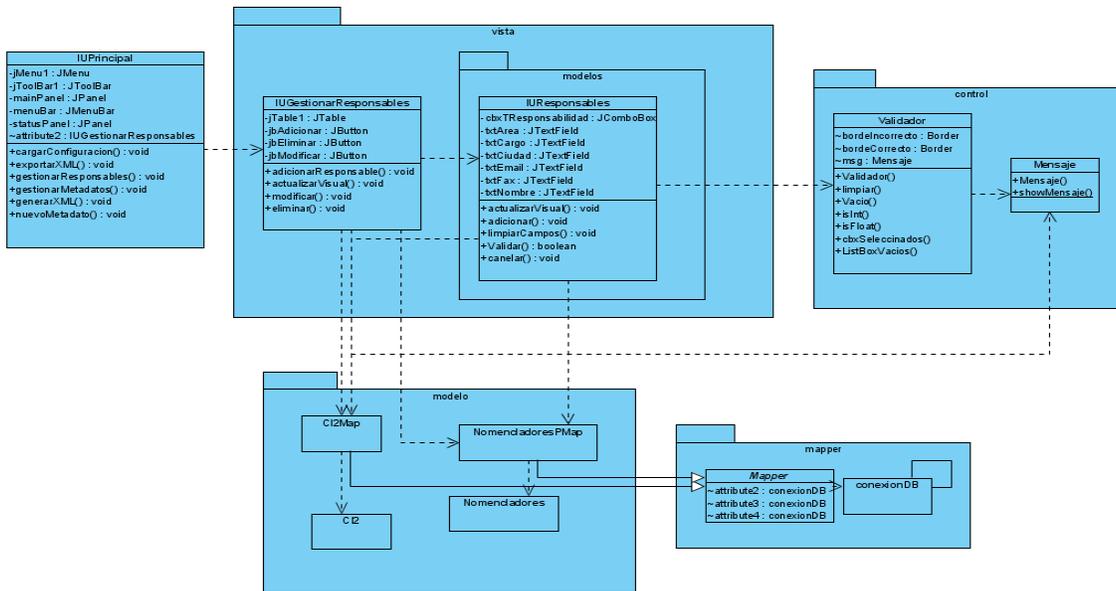


Figura 4: Diseño CU Gestionar datos de responsables.

4.3 Principios de diseño.

4.3.1 Estándares de interfaz de la aplicación.

Para lograr una interfaz de usuario amigable, atractiva y funcional para el usuario final, es necesario definir principios de diseño de interfaz de usuario, pues esta es una actividad de gran importancia dentro del diseño de un software. Se debe tener en cuenta que a partir de la aceptación que tengan los usuarios depende, en gran medida el éxito de nuestro sistema.

A continuación se definen los principios a tener en cuenta en el desarrollo de la interfaz de usuario de la aplicación:

Mostrar al usuario toda la información y herramientas necesarias para cada etapa en su trabajo.

- Brindar una interfaz sencilla, de manera tal que cualquier persona con un mínimo dominio de la computación pueda trabajar con la aplicación.

Capítulo 4: Solución propuesta

- Garantizar la legibilidad de manera que exista contraste de los colores de los textos con el fondo y el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.
- Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse que es correcta la opción seleccionada.
- Los mensajes mostrados al usuario deben ser concisos y de fácil comprensión.
- Menús y etiquetas de botones deben comenzar con la palabra más importante.
- Los eventos más importantes del sistema deben ser mostrados en una barra de estado.

4.3.2 Concepción general de la ayuda.

La herramienta a desarrollar contará con una ayuda general para facilitar la manipulación de la misma por parte de los usuarios finales. Esta se podrá encontrar en el menú principal para lograr una localización rápida. La estructura que se propone para organizar su contenido es la siguiente:

- Temas de Ayuda
- Estructura de la aplicación: Se describe la estructura de la aplicación.
Funcionalidades: Se explica de forma detallada cada una de las funcionalidades que brinda el sistema, a través del menú principal de la aplicación.
- Respuestas a Preguntas Frecuentes (FAQ).
- Acerca de: Se describen las características generales del software, atendiendo a: versión, desarrolladores, datos de contacto, año.

4.4 Diseño de la Base de Datos.

Mediante la persistencia se puede lograr que de los objetos se mantenga su valor en el tiempo y el espacio. Durante el diseño se identifican todas las clases persistentes, específicamente en el diagrama de clases persistentes; este diagrama muestra la estructura lógica de la base de datos mediante clases, traduciendo sus atributos a columnas de las tablas. A continuación se muestra en la figura 6 el diagrama de clases persistentes referentes a la aplicación a desarrollar y posteriormente se representa en la figura 7 el diagrama entidad relación de la base de datos que se genera del diagrama anterior.

Capítulo 4: Solución propuesta

4.4.1 Diagrama de clases persistentes.

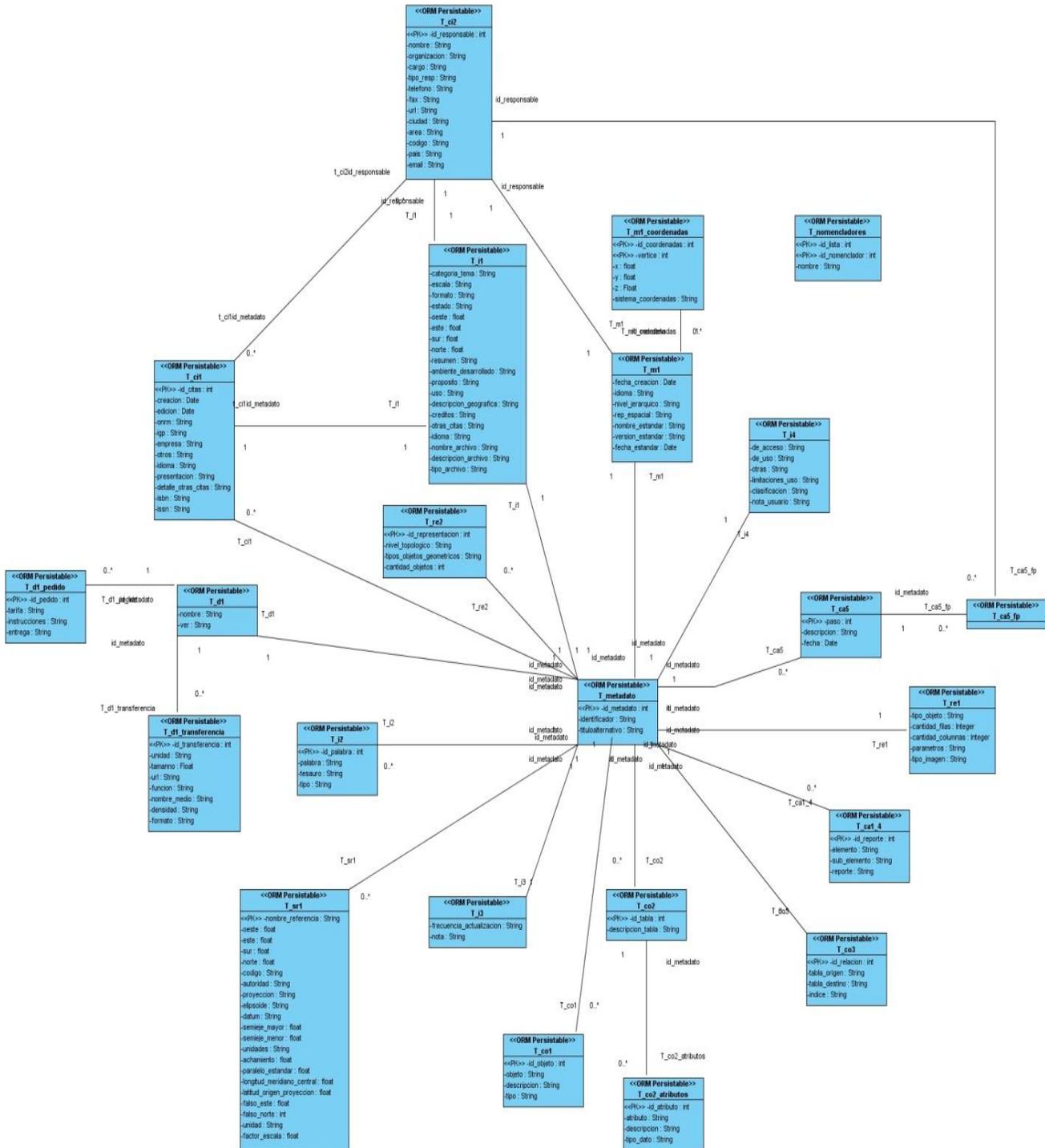


Figura 5 Diagrama de clases persistentes.

Capítulo 4: Solución propuesta

4.5 Diagrama de Despliegue.

En un diagrama de despliegue se muestra la distribución física de los objetos, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). A continuación se muestra el diagrama de despliegue de la solución propuesta:

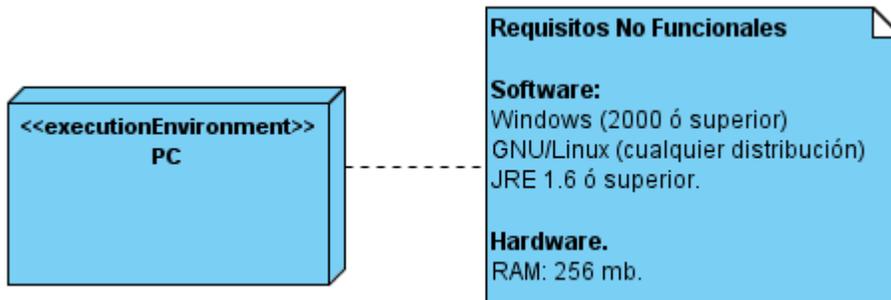


Figura 7 Diagrama de Despliegue.

4.6 Diagrama de Componentes.

Un componente es una parte física y reemplazable de un sistema.

Las clases representan abstracciones lógicas. Los componentes son elementos físicos del mundo real. Un componente es la implementación física de un conjunto de otros elementos lógicos, como clases y colaboraciones.

La relación que se establece entre componentes en un diagrama de componentes es de dependencia. Esto quiere decir que un componente necesita del otro para completar su definición.

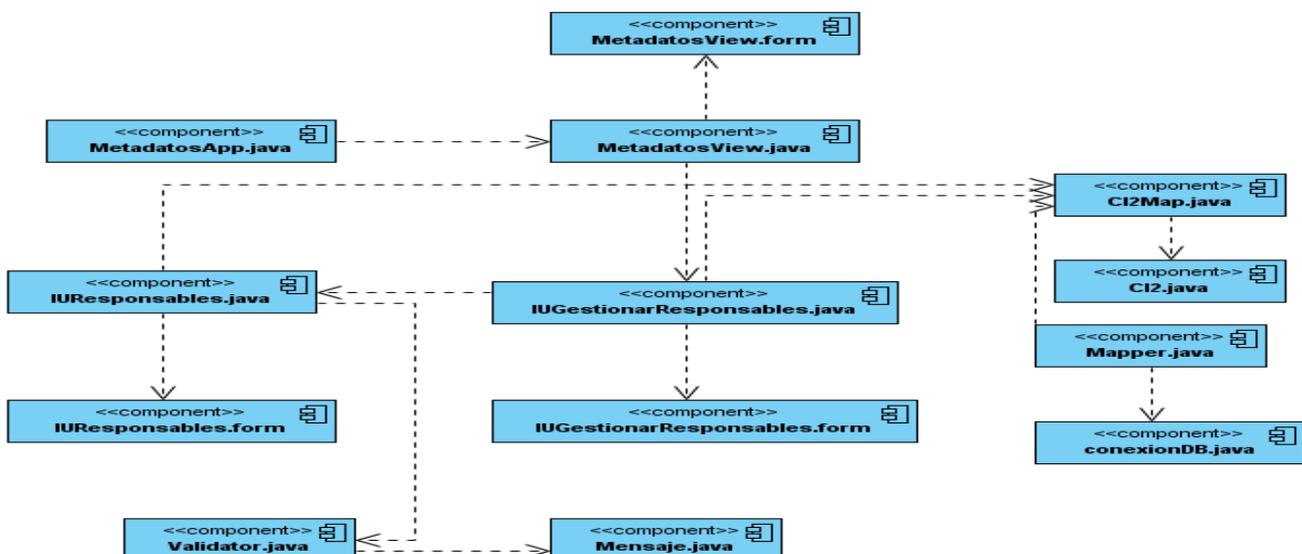


Figura 8 Diagrama de componentes del CU Gestionar datos de Responsable.

Capítulo 4: Solución propuesta

4.7 Pruebas al sistema propuesto.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requisitos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

4.7.1 Caso de uso Gestionar Datos de Responsables.

4.7.1.1 Descripción General

El CU se inicia cuando el especialista desea adicionar, eliminar o modificar los datos de un responsable.

4.7.1.2 Condiciones de Ejecución

La aplicación debe tener incluida la configuración para que el caso de uso funcione de forma correcta.

4.7.1.3 Secciones a probar

Mostrar Responsables

Adicionar Responsable

Modificar Responsable

Eliminar Responsable

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Mostrar responsables	EC 1.1 Mostrar tabla de responsables.	Se muestra una tabla con los responsables existentes.
SC 2: Adicionar responsable	EC 2.1 Se adiciona el responsable	El responsable se adiciona y se muestra en el listado de los responsables
	EC 2.2 El especialista no adicionar el responsable.	La interfaz de adicionar responsable es cerrada por el especialista.
	EC 2.3 Datos Incorrectos	Los datos que se necesitan para adicionar el responsable son

Capítulo 4: Solución propuesta

		incorrectos.
SC 3 Modificar responsable	EC 3.1 Se el responsable	El especialista modifica los datos del responsable y se muestra la tabla de los responsables existentes.
	EC 3.2 El especialista no modifica el responsable	El especialista cierra la interfaz de modificar responsable.
	EC 3.3 Datos incorrectos	Los datos necesarios para modificar el responsable son incorrectos.
SC 4: Eliminar responsable	EC 4.1 Se elimina un responsable	El especialista selecciona un responsable y luego lo elimina.
	EC 4.2 No se confirma la eliminación.	El especialista cancela la ventana de confirmación para eliminar un responsable.

Tabla 10 Secciones y escenarios CU Gestionar Datos de Responsables.

Descripción de las variables

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Responsable	Campo de texto	No	Número que identifica al responsable.
2	Nombre	Campo de texto	No	Nombre del responsable
3	Organización	Campo de texto	No	Organización a la que pertenece el responsable
4	Cargo	Campo de texto	No	Cargo del responsable
5	Tipo de responsabilidad	Lista desplegable	No	Listado de los tipos de responsabilidades.
6	Teléfono	Campo de texto	Sí	Número telefónico del responsable
7	Fax	Campo de texto	Sí	Número de fax del responsable
8	URL	Campo de texto	Sí	Dirección web asociada al

Capítulo 4: Solución propuesta

				responsable.
9	Ciudad	Campo de texto	No	Ciudad a la que pertenece el responsable
10	Área	Campo de texto	No	Área a la que pertenece el responsable
11	País	Campo de texto	No	País al que pertenece el responsable
12	Código	Campo de texto	Sí	Código asociado al país.
13	Email	Campo de texto	No	Dirección de correo electrónico del responsable
14	Aceptar	Botón	Sí	Permite adicionar o modificar el responsable
15	Cancelar	Botón	Sí	Permite Cancelar la inserción o modificación de un responsable

Tabla 11 Descripción de las variables CU Gestionar Datos de Responsables.

Capítulo 4: Solución propuesta

4.7.1.4 Matriz de datos.

4.7.1.4.1 SC 2: Adicionar responsable.

Escenario	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Respuesta del sistema	Resultado de la Prueba	Flujo Central
EC 2.1	V	V	V	V	V	V	V	V	V	V	V	V	V	V	N A	El sistema adiciona el responsable y muestra la lista de los responsables	Satisfactoria	Principal Gestionar Datos de Responsable Adicionar Responsable Aceptar
EC 2.2	N A	V	El sistema no adiciona el responsable y muestra la lista de los responsables	No Satisfactoria	<ul style="list-style-type: none"> Principal Gestionar Datos de Responsable Adicionar Responsable Cancelar 													
EC 2.3	I	V	V	V	V	V	V	V	V	V	V	V	V	V	N A	El sistema no adiciona el responsable y muestra un mensaje de error.	No Satisfactoria	<ul style="list-style-type: none"> Principal Gestionar Datos de Responsable
	V	I	V	V	V	V	V	V	V	V	V	V	V	V	N A			

Capítulo 4: Solución propuesta

V	V	I	V	V	V	V	V	V	V	V	V	V	V	V	N
V	V	V	I	V	V	V	V	V	V	V	V	V	V	V	A
V	V	V	V	I	V	V	V	V	V	V	V	V	V	V	N
V	V	V	V	V	I	V	V	V	V	V	V	V	V	V	A
V	V	V	V	V	V	I	V	V	V	V	V	V	V	V	N
V	V	V	V	V	V	V	I	V	V	V	V	V	V	V	A
V	V	V	V	V	V	V	V	I	V	V	V	V	V	V	N
V	V	V	V	V	V	V	V	V	I	V	V	V	V	V	A
V	V	V	V	V	V	V	V	V	V	I	V	V	V	V	N
V	V	V	V	V	V	V	V	V	V	V	I	V	V	V	A
V	V	V	V	V	V	V	V	V	V	V	V	I	V	V	N
V	V	V	V	V	V	V	V	V	V	V	V	I	V	V	A

- Adicionar Responsable
- Aceptar

Tabla 12 Matriz de datos Sección Adicionar Responsable.

Capítulo 4: Solución propuesta

4.8 Conclusiones parciales.

En el presente capítulo se muestran el diseño detallado de la aplicación, en el cual se hicieron uso de patrones de diseño y el diseño de la base de datos. Además se presenta como queda el sistema expresado en componentes de implementación y las pruebas realizadas al mismo, las que demuestran que la aplicación cumple con las funcionalidades previstas.

Conclusiones Generales

Conclusiones Generales.

En este punto se consideran cumplidos los objetivos trazados al tener desarrollado una herramienta para la creación de metadatos geológicos para la Oficina Nacional de Recursos Minerales. La misma cumple todos los requisitos planteados para su desarrollo por lo que se convierte en una herramienta cuyas funcionalidades básicas muestran los resultados esperados.

Se creó además una documentación técnica relacionada con la herramienta, en la misma se detallan todos los artefactos generados a partir de la aplicación de la Metodología de Desarrollo de Software Proceso Unificado de Desarrollo.

Este trabajo representa un aporte importante a la ONRM, ya que estandariza la creación de metadatos geológicos basado en la norma internacional ISO 19115 ajustada a la norma cubana, generando un fichero (XML) a partir de la implementación de la norma ISO 19139, permitiendo de esta manera mostrar el metadato geológico en un único formato a través de la herramienta GeoNetwork.

Recomendaciones.

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

- Presentar los resultados de la investigación en eventos de carácter científico, principalmente con tema de Geología.
- Hacer un estudio e implementación de la funcionalidad Importar XML, a partir de la norma cubana definida para los metadatos geológicos, basándose en la especificación técnica ISO 19139.

Referencias bibliográficas

Referencias bibliográficas

1. Definición.de. *Definición.de.* [En línea] [Citado el: 23 de Noviembre de 2009.]
<http://definicion.de/geologia/>.
2. Ministerio de la Industria Básica. *Ministerio de la Industria Básica.* [En línea] [Citado el: 2 de Diciembre de 2009.] http://www.cubagob.cu/des_eco/minbas.htm.
3. Oficina Nacional de Recursos Minerales. *Oficina Nacional de Recursos Minerales.* [En línea] [Citado el: 2 de diciembre de 2009.] <http://www.onrm.minbas.cu/>.
4. Facultad de Ciencias Geológicas. [En línea] [Citado el: 18 de 12 de 2009.]
<http://www.ucm.es/centros/webs/fgeo/>.
5. Oriente, Geominera. *Datos Geológicos.* Santiago de Cuba : Geominera Oriente, 2007.
6. Oliva, Rafael. *LOS METADATOS GEOGRÁFICOS: ACTUALIDAD Y ESTÁNDARES.* 2006.
7. Geoespatial Metadata Standards. [En línea] [Citado el: 10 de 12 de 2009.]
<http://www.fgdc.gov/metadata/geospatial-metadata-standards>.
8. Maganto, Alejandra Sánchez. *Instituto Geográfico Nacional Javier Nogueras.* Madrid : s.n., 2006.
NORMAS SOBRE METADATOS .
9. TICS - Tecnologías de la Información y las Comunicaciones. [En línea] [Citado el: 10 de Febrero de 2010.]
http://tics.org.ar/index.php?option=com_content&view=article&id=13:una-definicie-tics&catid=1:timas&Itemid=28.
10. Software Architecture. [En línea] 15 de 12 de 2009.
<http://www.sei.cmu.edu/architecture/start/community.cfm> .
11. UNADECODIGO. El paradigma Modelo Vista Controlador(Tutorial ROR II). *UNADECODIGO.* [En línea] [Citado el: 4 de Febrero de 2008.] <http://www.unadecodigo.com/2007/05/30/el-paradigma-modelo-vistacontrolador-tutorial-ror-ii/>.
12. Glosario. [En línea] [Citado el: 10 de Febrero de 2010.] <http://tecnologia.glosario.net/terminos-tecnicos-internet/uml-1655.html>.
13. Metodologías de Desarrollo de Software. [En línea] [Citado el: 25 de Febrero de 2008.]
http://static.scribd.com/docs/dsmodb48dz9x5.swf?INITIAL_VIEW=EW=width.
14. Metodologías tradicionales vs. Metodologías ágiles. . [En línea] [Citado el: 4 de Marzo de 2010.]
http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.1515.
15. Microsoft Foundation Software. [En línea] [Citado el: 4 de Marzo de 2010.]
<http://caraujo334.blogspot.es>.
16. Lenguaje de programación. [En línea] [Citado el: 26 de Febrero de 2010.]
<http://es.kioskea.net/contents/langages/langages.php>.

Referencias bibliográficas

17. Que es Java. [En línea] [Citado el: 24 de Febrero de 2010.] http://java.ciberaula.com/articulo/que_es_java.
18. Curos Introdutorio de Java. [En línea] [Citado el: 21 de Febrero de 2010.] <http://www.mailxmail.com/curso-java/caracteristicas-lenguaje-java-3>.
19. INtroducción al lenguaje C# y al Framework .NET. [En línea] [Citado el: 4 de Marzo de 2010.] <http://msdn.microsoft.com/es-es/library>.
20. Introducción al lenguaje C. [En línea] [Citado el: 2 de Marzo de 2010.] <http://www.articulandia.com/premium/article.php/12-03-2007El-Lenguaje-C.htm>.
21. Lenguaje C++. [En línea] [Citado el: 2 de Marzo de 2010.] http://www.zator.com/Cpp/E1_2.htm.
22. *Herramienta CASE*. [En línea] [Citado el: 12 de Febrero de 2010.] <://www.slideshare.net/guest15d257/herramientas-case-508428>.
23. CASE. Ventajas e inconvenientes. *Consejo Superior de Administración Electrónica*. [En línea] [Citado el: 19 de Enero de 2008.] <http://www.csi.map.es/csi/silice/Global75.html>.
24. Visual Paradigm Internacional. [En línea] [Citado el: 10 de Febrero de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%.
25. Introducción a Rational Rose. [En línea] [Citado el: 12 de Marzo de 2010.] <http://sistemaronald.blogspot.com/2008/08/introduccion-rational-rose.html>.
26. Entornos de Desarrollo Integrados. [En línea] [Citado el: 8 de Febrero de 2010.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
27. Bienvenido a NetBeans. [En línea] [Citado el: 20 de Febrero de 2010.] http://netbeans.org/index_es.html.
28. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El proceso unificado de desarrollo de sosftware*. Madrid : Addison Wesley 1998. ISBN/ISSN.
29. Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. New York : Oxford University Press, 1977.
30. Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford. *Patterns of Enterprise Application Architecture*. Boston : Addison Wesley 2002.

Bibliografía

1. Definición.de. *Definición.de*. [En línea] [Citado el: 23 de Noviembre de 2009.] <http://definicion.de/geologia/>.
2. Ministerio de la Industria Básica. *Ministerio de la Industria Básica*. [En línea] [Citado el: 2 de Diciembre de 2009.] http://www.cubagob.cu/des_eco/minbas.htm.
3. Oficina Nacional de Recursos Minerales. *Oficina Nacional de Recursos Minerales*. [En línea] [Citado el: 2 de diciembre de 2009.] <http://www.onrm.minbas.cu/>.
4. Facultad de Ciencias Geológicas. [En línea] [Citado el: 18 de 12 de 2009.] <http://www.ucm.es/centros/webs/fgeo/>.
5. Oriente, Geominera. *Datos Geológicos*. Santiago de Cuba : Geominera Oriente, 2007.
6. Oliva, Rafael. *LOS METADATOS GEOGRÁFICOS: ACTUALIDAD Y ESTÁNDARES*. 2006.
7. Geoespatial Metadata Standards. [En línea] [Citado el: 10 de 12 de 2009.] <http://www.fgdc.gov/metadata/geospatial-metadata-standards>.
8. Maganto, Alejandra Sánchez. *Instituto Geográfico Nacional Javier Nogueras*. Madrid : s.n., 2006. NORMAS SOBRE METADATOS .
9. TICS - Tecnologías de la Información y las Comunicaciones. [En línea] [Citado el: 10 de Febrero de 2010.] http://tics.org.ar/index.php?option=com_content&view=article&id=13:una-definicie-tics&catid=1:timas&Itemid=28.
10. Software Architecture. [En línea] 15 de 12 de 2009. <http://www.sei.cmu.edu/architecture/start/community.cfm> .
11. UNADECODIGO. El paradigma Modelo Vista Controlador(Tutorial ROR II). *UNADECODIGO*. [En línea] [Citado el: 4 de Febrero de 2008.] <http://www.unadecodigo.com/2007/05/30/el-paradigma-modelo-vistacontrolador-tutorial-ror-ii/>.
12. Glosario. [En línea] [Citado el: 10 de Febrero de 2010.] <http://tecnologia.glosario.net/terminos-tecnicos-internet/uml-1655.html>.
13. Metodologías de Desarrollo de Software. [En línea] [Citado el: 25 de Febrero de 2008.] http://static.scribd.com/docs/dsmodb48dz9x5.swf?INITIAL_VIEW=EW=width.
14. Metodologías tradicionales vs. Metodologías ágiles. . [En línea] [Citado el: 4 de Marzo de 2010.] http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.1515.
15. Microsoft Foundation Software. [En línea] [Citado el: 4 de Marzo de 2010.] <http://caraujo334.blogspot.es>.
16. Lenguaje de programación. [En línea] [Citado el: 26 de Febrero de 2010.] <http://es.kioskea.net/contents/langages/langages.php>.

Glosario de términos

17. Que es Java. [En línea] [Citado el: 24 de Febrero de 2010.] http://java.ciberaula.com/articulo/que_es_java.
18. Curos Introdutorio de Java. [En línea] [Citado el: 21 de Febrero de 2010.] <http://www.mailxmail.com/curso-java/caracteristicas-lenguaje-java-3>.
19. INtroducción al lenguaje C# y al Framework .NET. [En línea] [Citado el: 4 de Marzo de 2010.] <http://msdn.microsoft.com/es-es/library>.
20. Introducción al lenguaje C. [En línea] [Citado el: 2 de Marzo de 2010.] <http://www.articulandia.com/premium/article.php/12-03-2007El-Lenguaje-C.htm>.
21. Lenguaje C++. [En línea] [Citado el: 2 de Marzo de 2010.] http://www.zator.com/Cpp/E1_2.htm.
22. *Herramienta CASE*. [En línea] [Citado el: 12 de Febrero de 2010.] <://www.slideshare.net/guest15d257/herramientas-case-508428>.
23. CASE. Ventajas e inconvenientes. *Consejo Superior de Administración Electrónica*. [En línea] [Citado el: 19 de Enero de 2008.] <http://www.csi.map.es/csi/silice/Global75.html>.
24. Visual Paradigm Internacional. [En línea] [Citado el: 10 de Febrero de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%.
25. Introducción a Rational Rose. [En línea] [Citado el: 12 de Marzo de 2010.] <http://sistemaronald.blogspot.com/2008/08/introduccion-rational-rose.html>.
26. Entornos de Desarrollo Integrados. [En línea] [Citado el: 8 de Febrero de 2010.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
27. Bienvenido a NetBeans. [En línea] [Citado el: 20 de Febrero de 2010.] http://netbeans.org/index_es.html.
28. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El proceso unificado de desarrollo de sosftware*. Madrid : Addison Wesley 1998. ISBN/ISSN.
29. Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. New York : Oxford University Press, 1977.
30. Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford. *Patterns of Enterprise Application Architecture*. Boston : Addison Wesley 2002.
31. Sánchez, Alejandra y Ballari, Daniella. *Normas sobre los metadatos (ISO 19115, ISO 19115-2 , ISO19932)*. 2008.
32. Quesada, Orozco y Santos, Eduardo. *Los Metadatos Geográficos: Actualidad y Estándares*. Ciudad de la Habana : s.n., 2006.
33. Verdecie, Lisbet O. y Muñoz Suarez, José L. *Subsistema para la consulta de metadatos geologicos en la Oficina Nacional de Recursos Minerales*. Ciudad de la Habana : s.n., 2008.

Glosario de términos.

Automatizar: Se le denomina así a cualquier tarea realizada por máquinas en lugar de personas. Es la sustitución de procedimientos manuales por sistemas de cómputo.

Applet: es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

Bytecodes: es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto

COBOL8: acrónimo de Common Business -Oriented Language, Lenguaje Común Orientado a Negocios) fue creado en el año 1960.

Data-Mapper: Patrón de diseño vinculado a lo referente a acceso a datos.

Depurador: (en inglés, debugger), es un programa que permite depurar o limpiar los errores de otro programa informático.

Framework: es un esquema, esqueleto o patrón para el desarrollo y/o la implementación de una aplicación.

GPL: Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

HTML: es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet.

ISO: Organización Internacional para la Estandarización o ISO, cuyo nombre en inglés es International Organization for Standardization.

JRE: Java Runtime Environment, es un conjunto de utilidades que permite la ejecución de programas Java.

Multi-plataforma: sistema informático que corre sobre varios sistemas operativos, sin prescindir de ninguna de sus funcionalidades

Glosario de términos

Open Source: Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente

Pascal: Pascal es un lenguaje de programación, creado con el objetivo de facilitar el aprendizaje de programación en estudiantes, utilizando la programación estructurada y estructuración de datos.

RAM: memoria de acceso aleatorio (en inglés: random - access memory) es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

ResultSet: tipo de dato q guarda los datos que retorna una consulta en HSQLDB.

SOA: tipo de arquitectura, en este caso la arquitectura orientada a servicio.

Socket: Método de comunicación entre el servidor y el cliente, o bien entre programas dentro el mismo ordenador.

Sun Microsystems: empresa informática fabricante de semiconductores y software.

Tupla: representación de una fila en una de las tablas que se está almacenando datos.

XML: metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específico

W3C: World Wide Web Consortium, es un consorcio internacional que produce recomendaciones para la World Wide Web.