

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



*Herramienta para la captura de datos de campo del Inventario de
Minerales Sólidos perteneciente al Sistema de Gestión de Datos
Geológicos.*

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

AUTOR:

Yamil Moya Vera

TUTOR: *Ing. Dagoberto Antonio Suarez Morales*

CO-TUTOR: *Ing. Eddy Dangel Quezada Rodríguez*

Cuidad de la Habana, 5 de Julio del 2010.

Año del 52 Aniversario de la Revolución.

*A Nilda, por guiarme por el camino de las ciencias, por brindarme su amor
incondicional en cada momento.*

Agradecimientos

A mi familia, en especial a todo aquel que ha brindado su granito de arena.

A mis padres, por regalarme la vida, en especial a Felicita, Mario, Nilda, Juana y Elisa por cuidarme en cada momento, por el amor sin límites, el sacrificio, las preocupaciones y la confianza.

A mi abuela, por guiarme por el camino de las ciencias.

A mi hermana Yanet por tanto amor.

A mis hermanos y mi padre, responsables y sacrificados, Erick, Harold y Esteban por tanto amor.

A los tíos, Magalís, Santiago, José Ramón por brindarme su ayuda en cada momento.

A mi novia Lily por tanto apoyo, consejos, amor incondicional y demostrar que de nada vale vivir si no se siente estar vivo.

A los amigos, en especial a aquellos como hermanos, Raúl, Cesar, Alain y Mirabal.

A Yalína y Alejandro, por brindarme su ayuda y hospitalidad durante estos cinco años.

Al los ingenieros Dagoberto A. y Eddy Dangel por tutorar, por los consejos y las ideas.

A todos aquellos que aportaron ideas y consejos para llevar a cabo esta obra. Gracias a Alain Companioni, Tony, Yudiel, Alberto, Reinaldo, Héctor y Rocny.

Al grupo de trabajo de SQDG en general.

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al proyecto SGD G de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor: Yamil Moya Vera _____

Tutor: Dagoberto A. Suarez Morales

Co-Tutor: Eddy D. Quezada Rodríguez

Datos de Contacto

Síntesis del Tutor

Ing. Dagoberto Antonio Suarez Morales

Profesión: Ingeniero en Ciencias Informáticas

Categoría docente: Adiestrado

Año de graduado: 2008

Síntesis del Co-Tutor

Ing. Eddy Dangel Quezada Rodríguez

Profesión: Ingeniero en Ciencias Informáticas

Categoría docente: Adiestrado

Año de graduado: 2008

Opiniones y Avales

Opinión del Tutor

Resumen

En el presente trabajo se elabora un estudio sobre los diferentes procesos que ocurren en la actualidad en la captura de datos de campo en los concesionarios para la realización del Inventario Nacional de Minerales Sólidos perteneciente a la Oficina Nacional de Recursos Minerales (ONRM). Se identifican las principales deficiencias en la aplicación que le da solución parcialmente a dichos procesos. Se traza como solución, la redacción de la documentación necesaria, así como el desarrollo del sistema, para de esta manera desarrollar una aplicación que preste solución a las deficiencias existentes actualmente. El documento colecciona los resultados de la investigación desarrollada. Se identifica y analiza exhaustivamente la situación problemática y el dominio donde ocurren los principales procesos del negocio. Es realizado el análisis de las diferentes tendencias y tecnologías actuales candidatas para el modelado y desarrollo del sistema. Posteriormente se enumeran las distintas funcionalidades que debe poseer el sistema y se realiza su diseño para determinar cómo debe quedar estructurada para su mejor desarrollo. Se realiza el diseño del sistema propuesto basado en los requerimientos planteados anteriormente, se modela su implementación y finalmente se procede con la implementación del sistema y las pruebas necesarias para verificar la vitalidad del programa.

PALABRAS CLAVES:

Captura de Datos de Campo, Concesión, Concesionario, Inventario, Recurso Mineral, Reserva Mineral.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.2.1 Captura de datos de campo.....	6
1.2.2 Concesión.....	7
1.2.3 Concesionario.....	7
1.2.4 Inventario.....	8
1.2.5 Recurso Mineral.....	8
1.2.6 Reserva Mineral.....	10
1.3 Objeto de Estudio.....	10
1.3.1 Descripción General.....	10
1.3.2 Descripción actual del dominio del problema.....	11
1.3.3 Situación Problemática.....	13
1.4 Análisis de otras soluciones existentes.....	13
1.5 Conclusiones parciales.....	14
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR.	15
2.1 Introducción.....	15
2.2 Software Libre vs. Software privativo.....	15
2.2.1 Software privativo.....	16
2.2.2 Software libre.....	16
2.3 Lenguajes de programación.....	17

2.3.1	Lenguaje C++.....	17
2.3.2	Lenguaje Java.....	18
2.3.3	Criterios de Comparación.....	18
2.4	Fundamentación del lenguaje de programación seleccionado.....	21
2.5	El Lenguaje Unificado de Modelado (UML 2.0) como soporte de la modelación de la solución propuesta.....	21
2.6	Metodologías para el desarrollo del sistema.....	23
2.6.1	Programación Extrema (XP).....	23
2.6.2	RUP.....	24
2.6.3	Fundamentación de la metodología de desarrollo escogida.....	26
2.7	Lenguaje XML.....	26
2.7.1	¿Por qué se utilizó XML como fichero para exportar?.....	26
2.8	Arquitectura Modelo Vista Controlador (MVC).....	27
2.8.1	¿Por qué se utilizó como arquitectura MVC?.....	28
2.9	Propuesta de Herramientas para el desarrollo del sistema.....	28
2.9.1	NetBeans 6.8 M II como IDE para la implementación del sistema.....	28
2.9.2	Visual Paradigm como herramienta CASE para el modelado del sistema.....	29
2.9.3	HSQLDB como sistema gestor de base de datos.....	30
2.10	Sistema operativo para el desarrollo de la aplicación.....	31
2.11	Framework para el desarrollo del sistema.....	32
2.11.1	Swing.....	32
2.12	Conclusiones parciales.....	33
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN.....		34

3.1	Introducción.....	34
3.2	Modelo de Negocio.....	34
3.2.1	Actores y trabajadores del negocio.....	34
3.2.2	Procesos del negocio.	35
3.2.3	Diagrama de Casos de Uso del Negocio.	36
3.2.4	Descripción textual de los Casos de Uso del Negocio.	36
3.2.5	Diagrama de Objetos del Negocio.	38
3.3	Requisitos funcionales y no funcionales.....	39
3.3.1	Requisitos funcionales.....	39
3.3.2	Requisitos no funcionales.	40
3.4	Descripción del sistema propuesto.	41
3.4.1	Descripción de los actores.....	41
3.4.2	Casos de uso del sistema.....	42
3.5	Conclusiones parciales.....	49
CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....		50
4.1	Introducción.....	50
4.2	Modelo de diseño.	50
4.2.1	Diagramas de diseño.....	50
4.3	Principios de diseño.....	51
4.3.1	Estándares de la interfaz de la aplicación.....	52
4.3.2	Concepción general de la ayuda.....	53
4.4	Diseño de la base de datos.....	54
4.5	Modelo de despliegue.....	55

4.6	Modelo de Implementación.....	55
4.7	Pruebas del sistema propuesto.....	57
4.8	Conclusiones Parciales.....	59
CONCLUSIONES GENERALES.....		60
TRABAJOS REFERENCIADOS		61
BIBLIOGRAFÍA.....		63
GLOSARIO DE TÉRMINOS		65
ANEXOS		67

ÍNDICE DE TABLAS

Tabla 1 Recursos identificados, no identificados y sus clasificaciones.....	9
Tabla 2 Comparación entre lenguajes de programación.	21
Tabla 3 Actores del Negocio y su Descripción.	34
Tabla 4 Trabajadores del Negocio y su Descripción.	35
Tabla 5 Descripción textual del Caso de Uso Registrar Datos de Campo.	36
Tabla 6 Descripción textual del Caso de Uso Entregar Informes Geológicos.	37
Tabla 7 Actores del Sistema y su Descripción.	42
Tabla 8 Descripción textual del Caso del Uso Gestionar Recursos y Reservas.	43
Tabla 9 Descripción textual del Caso de Uso Exportar Informe Digital.....	47

ÍNDICE DE IMÁGENES

Imagen 1 Curso de los documentos en la ONRM.	12
Imagen 2 Jerarquía de los diagramas UML 2.0.....	22
Imagen 3 Los cinco flujos de trabajo – requisitos, análisis, diseño, implementación y prueba- tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición.	25
Imagen 4 Arquitectura Modelo Vista Controlador (MVC).....	27
Imagen 5 Diagrama de Casos de Uso del Negocio.....	36
Imagen 6 Diagrama de Objetos del Negocio.....	39
Imagen 7 Diagrama de Casos de Uso del Sistema.....	43
Imagen 8 Diagrama de Casos de Usos del Sistema paquete Gestionar Datos Geológicos.	43
Imagen 9 Diagrama de Casos de Uso del Sistema paquete Generar Informes.....	47
Imagen 10 Diagrama de Clases del Diseño del Caso de Uso Gestionar Informe Geológico.	51
Imagen 11 Diagrama Entidad - Relación.....	54
Imagen 12 Diagrama de Despliegue.....	55
Imagen 13 Diagrama de Componentes Modelo Genérico.....	56
Imagen 14 Diagrama de Componentes del Caso de Uso Generar Informe Recursos y Reservas.	56
Imagen 15 de Actividades Caso de Uso Entregar Informes Geológicos.....	67

INTRODUCCIÓN

En la actualidad la civilización moderna, sus patrones de desarrollo y estilos de vida, se sustentan en el uso de una gama sumamente amplia de recursos minerales existentes en la corteza terrestre y en los fondos marinos, que son transformados, gracias a la tecnología, en materiales susceptibles a ser usados por la sociedad. Desde tiempos inmemorables se muestra que no es posible prescindir de la explotación de los recursos minerales, llámese recursos minerales a todos los metales, minerales, rocas, e hidrocarburos, que pueden ser utilizados por el hombre y que existen en el suelo y subsuelo.

En la actualidad, el organismo encargado de controlar las extracciones de los recursos minerales, entre otras tareas, es el Ministerio de la Industria Básica, específicamente una de sus entidades adscritas, la ONRM (MINBAS, 2009), entidad rectora a nivel nacional que garantiza la racional explotación y utilización de los recursos minerales e implementa el marco jurídico para el desarrollo y control de la geología, la minería y el petróleo (MINBAS, 2009). La ONRM es creada por la Ley de Minas (Capítulo V, sección primera, artículo 14), aprobada por la Asamblea Nacional del Poder Popular en 1995 (Asamblea, 1995). Una de las funciones fundamentales de la ONRM es el mantenimiento de las estadísticas de la información geológica almacenada en la oficina.

Hoy en día, la ONRM se encuentra realizando grandes esfuerzos por informatizar y automatizar todos sus servicios auxiliándose en las nuevas tecnologías de la Informática y las Comunicaciones. Como estrategia para su desarrollo comenzó a realizar colaboraciones con la Universidad de las Ciencias Informáticas (UCI), hecho que concluyó con el acuerdo de la realización del análisis y desarrollo de diferentes sistemas por parte de la universidad que cumplirán con todas las expectativas mencionadas anteriormente y garantizarán un ágil y sencillo manejo de la información geológica.

El proceso de captura de datos, en los concesionarios, se realiza en el quehacer diario para el Inventario Nacional de Minerales Sólidos, en la ONRM, con el objetivo de que la dirección de la Revolución posea toda la información sobre el estado de los recursos

minerales en el país, controlando y garantizando el uso racional de los mismos y, de esta manera, trazar nuevas estrategias para garantizar una mayor sostenibilidad científica, técnica, económica, político y social del país.

Para la captura de datos del Inventario Nacional de Minerales Sólidos en la ONRM, existen numerosas deficiencias que generalmente hacen muy engorroso el trabajo de los geólogos dentro de la oficina. Actualmente realizan la gestión de toda la actividad minera de cada uno de los concesionarios utilizando como sistema gestor de datos (Microsoft Access 97), este gestor no cumple con las expectativas necesarias y es limitado en el volumen de datos a soportar.

Cada año, la ONRM recibe de cada concesionario existente en el país, un reporte de la actividad minera realizada. Este reporte consiste en dos informes principales por cada concesión, el Informe Inventario (balance de recursos y reservas) e Informe Geológico, acumulándose un gran volumen de información a revisar, corregir e insertar en el sistema en un período de tiempo limitado, provocando que la revisión e inserción se realice de manera lenta y la cantidad de información acumulada dificulte su digitalización.

Para la inserción, modificación y eliminación de la información, es necesaria la utilización de un conjunto de software de naturaleza propietaria, su seguridad y acceso cuentan con notables deficiencias. Debido al volumen limitado de información soportado por Access 97, puede en cualquier momento existir una gran pérdida de información, muy sensible para nuestro gobierno. Por todo lo anteriormente mencionado, las condiciones de trabajo que existen no son las eficientes, ya que cuentan con una tecnología informática que no se adapta a las condiciones actuales de la oficina y conduce a obtener resultados que en muchas ocasiones no suelen ser los esperados.

Las deficiencias anteriormente planteadas, conducen al siguiente **problema a resolver**:
¿Cómo agilizar los procesos de captura de datos de campo del módulo Inventario Nacional de Minerales Sólidos que forma parte del Sistema de Gestión de Datos Geológicos?

Para darle solución al problema planteado se define como **objeto de estudio** los procesos de captura de datos de campo del módulo Inventario Nacional de Minerales Sólidos del Sistema de Gestión de Datos Geológicos y como **campo de acción** la informatización de los procesos de captura de datos de campo del Inventario Nacional de Minerales Sólidos del Sistema de Gestión de Datos Geológicos.

El **objetivo general** es desarrollar una aplicación que permita la captura de los datos de campo del módulo Inventario Nacional de Minerales Sólidos que forma parte del Sistema de Gestión de Datos Geológicos.

La **idea a defender** es: El desarrollo de una herramienta informática permitirá agilizar los procesos de captura de datos de campo del módulo Inventario Nacional de Minerales Sólidos que forma parte del Sistema de Gestión de Datos Geológicos.

Para el cumplimiento de los objetivos planteados previamente se trazan las siguientes **tareas investigativas**:

- Caracterizar el proceso de captura de datos de campo para el Inventario Nacional de Minerales Sólidos 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 Inventario Nacional de Minerales Sólidos.
- Especificar los requisitos funcionales del software.
- Elaborar el Diagrama de Casos de Uso del Sistema del módulo Inventario Nacional de Minerales Sólidos.
- Elaborar los diagramas de clase del diseño del módulo Inventario Nacional de Minerales Sólidos.
- Elaborar el diagrama de implementación.

- Implementar los CU definidos.
- Desarrollar los Casos de Prueba que certifiquen la veracidad de los algoritmos empleados.

Los **métodos científicos** de la investigación utilizados fueron:

Teóricos:

- Analítico-Sintético: Con la realización de un análisis de los documentos y la bibliografía se obtuvo una síntesis detallada de los mismos.
- Análisis Histórico Lógico: Este método permitió el análisis histórico y lógico del proceso de captura de datos de campo del Inventario Nacional de Minerales Sólidos pertenecientes al Sistema de Gestión de Datos Geológicos.
- Modelación: Se modeló a través de diagramas los distintos procesos y actividades que se desarrollan en la elaboración del sistema.

Empíricos:

- Observación: En las visitas realizadas a la ONRM se observó cómo se realizaban los procesos de captura de datos de campo del Inventario Nacional de Minerales Sólidos alcanzándose así una mejor comprensión de estos mecanismos.
- Entrevista: En las entrevistas realizadas a los diferentes trabajadores de la ONRM encargados de la captura de datos de campo del Inventario Nacional de Minerales Sólidos se logró obtener las principales deficiencias que dan origen al problema científico planteado.

Resultado Esperado:

- Obtener una aplicación con la cual disminuir considerablemente el esfuerzo respecto al tiempo en el cual es realizado el Inventario Nacional de Minerales Sólidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

Mantener un estricto control de los recursos minerales existentes en el territorio cubano resulta de vital importancia para nuestro país. Para esto se realiza incesablemente la captura de datos geológicos en cada una de las concesiones de la nación, informando constantemente toda su actividad minera para la confección del Inventario Nacional de Minerales Sólidos, el cual es llevado a cabo por la ONRM, adscrita actualmente al Ministerio de la Industria Básica.

En el presente capítulo se abordarán los fundamentos generales con las bases conceptuales que permitirán el entendimiento del problema. Se describirá eficazmente como ocurren los procesos de captura de datos de campo, vital proceso para la economía cubana y la manera en la cual se manifiestan estos procesos en la ONRM. Se realizará una descripción del objeto de estudio de la investigación al igual que la descripción del dominio y la situación problemática de la misma.

1.2 Conceptos asociados al dominio del problema.

Existen un conjunto de conceptos que están muy relacionados con el problema planteado anteriormente los cuales son descritos a continuación (captura de datos de campo, concesión, concesionario, recurso mineral, reserva mineral).

1.2.1 Captura de datos de campo.

El uso racional de los recursos minerales es una actividad de dirección, decisión y ejecución de la minería nacional encaminada a asegurar la más provechosa utilización de las riquezas minerales del país, llámese datos de campo a la información que se relaciona con: (MINBAS, 1999)

- La investigación geológica, dirigida al descubrimiento, reconocimiento y utilización de los recursos minerales, incluyendo el aprovechamiento óptimo de los recursos materiales, utilizados para realizar la misma.

- La construcción de minas y canteras.
- La explotación y procesamiento de los minerales.
- La protección de los recursos minerales.
- La mitigación del impacto ambiental.
- El abandono de recursos, reservas y el cierre de minas (MINBAS, 1999).

La captura de datos de campo no es más que el registro de toda esta actividad minera realizada por los concesionarios mediante dos informes principales, Informe Geológico e Informe Inventario utilizando un estricto formato (Instrucción del Balance) (MINBAS, 1999).

1.2.2 Concesión.

En numerosas ocasiones encontramos en el proceso de captura de datos geológicos la terminología concesión, lo cual resulta de vital importancia conocer su significado, a continuación se hace referencia al artículo de la *Ley de Minas* (Asamblea, 1995) que lo conceptualiza:

ARTÍCULO 17 Ley de Minas.- A los efectos de esta Ley se entiende por Concesión Minera, que en lo sucesivo se denomina genéricamente Concesión, la relación jurídica nacida de un acto gubernativo unilateral por el que se otorga temporalmente a una persona natural o jurídica el derecho de realizar actividades mineras, bajo las condiciones y con todos los derechos y obligaciones que esta Ley y su Reglamento determinan. Todos los recursos minerales que se relacionan en el artículo 13 de la presente Ley son concesibles, sin perjuicio de la declaración de reservas exclusivas de determinados minerales por parte del Estado (Asamblea, 1995).

1.2.3 Concesionario.

Resulta también de vital importancia conocer el significado de concesionario, a continuación se hace referencia al artículo de la *Ley de Minas* (Asamblea, 1995) que lo conceptualiza:

ARTÍCULO 20 Ley de Minas. - Son concesionarios, a los efectos de esta Ley, las personas naturales o jurídicas, debidamente autorizadas para el ejercicio de una o varias fases de la actividad minera por el correspondiente título (Asamblea, 1995).

1.2.4 Inventario.

Inventario según la *Real Academia Española*:

Aquel en que periódicamente se han de hacer constar todos los bienes y derechos del activo y todas las deudas y obligaciones del pasivo de cada comerciante, persona natural o jurídica y balance general de su giro (Española).

El Inventario Nacional de Minerales Sólidos, también llamado Balance de Recursos y Reservas Minerales de la Nación se elabora anualmente, con el objetivo de tener toda la información sobre el estado de las reservas y recursos minerales controlando y garantizando el uso racional de las mismas; ésta es, entre otras, una de las funciones de la ONRM (Minerales).

1.2.5 Recurso Mineral.

Concentración de minerales o elementos útiles sólidos que existen en la corteza terrestre, tanto en la superficie como en profundidad, cuyas características hacen posible su extracción económicamente favorable en las condiciones actuales o futuras. Se clasifican en **Identificado** y **no Identificado** (MINBAS).

Es el recurso mineral **identificado** “in situ” del cual pueden obtenerse minerales útiles valiosos, cuya ubicación, morfología, calidad y cantidad se conocen o se han estimado por los datos geológicos. El Recurso Mineral Identificado se clasifica en: Medido, Indicado e Inferido.

El recurso **no Identificado** es aquel recurso supuesto o intuido, que se estima con un alto grado de incertidumbre mediante la extrapolación geológica, premisas, criterios (geofísicos, geoquímicos, etc.) o por los cálculos estadísticos. La estimación debe

realizarse independiente del resto de los recursos. Se subdividen en: Hipotético y Especulativo.

Tabla 1 Recursos identificados, no identificados y sus clasificaciones.

Recursos Identificados	
Medido	Recurso estimado sobre la base de datos geológicos obtenidos mediante perforaciones, laboreos mineros y en afloramientos, que están ubicados a distancias suficientemente cercanas que confirmen su continuidad. La cantidad, interpretación y confiabilidad de los datos deben permitir una clara determinación de la forma, tamaño y calidad del cuerpo mineral (MINBAS).
Indicado	Recurso estimado sobre la base de datos geológicos obtenidos mediante perforaciones, laboreos mineros y en afloramientos, que están ubicados a distancias suficientemente cercanas pero más espaciados que en el Medido y confirman de forma razonable su continuidad. La cantidad, interpretación y confiabilidad de los datos deben permitir una determinación adecuada de la forma, tamaño y calidad del cuerpo mineral (MINBAS).
Inferido	Recurso estimado sobre la base de datos geológicos obtenidos mediante perforaciones, laboreos mineros y en afloramientos pero la cantidad de datos y su confiabilidad no es suficiente para realizar una interpretación confiable del marco geológico ni para predecir la continuidad del mineral (MINBAS).
Recursos No identificados	
Hipotético	Recurso que puede ser descubierto mediante la ampliación del área del Recurso Identificado y aquel cuya existencia en la región puede ser esperada razonablemente, en correspondencia con las condiciones geológicas existentes (MINBAS).

Especulativo	Recurso no descubierto aún pero que puede hacerse evidente por la existencia de premisas o criterios geológicos favorables, en: cuencas, campos minerales, regiones de nuevos yacimientos o depósitos que aún no han sido reconocidos por su potencial económico (MINBAS).
--------------	--

1.2.6 Reserva Mineral.

La reserva mineral es la parte o totalidad del recurso mineral medido o indicado que puede ser extraído con utilidad económica, considerando las pérdidas y dilución, según las condiciones reales asumidas en el Estudio de Factibilidad o Pre Factibilidad Económica (MINBAS).

Los estimados de reservas minerales provienen de los recursos, en cuya estimación se han considerado criterios mineros, tecnológicos, medioambientales, económicos, legales, sociales y gubernamentales.

La reserva mineral se clasifica en: **probada** y **probable**.

- La reserva mineral **probada** es el recurso mineral medido, cuya viabilidad de extracción en el momento de su cálculo, está soportado en un estudio de factibilidad (MINBAS).
- La reserva mineral **probable** es el recurso mineral indicado, cuya viabilidad de extracción en el momento de su cálculo, está soportado en un Estudio de Pre factibilidad o Factibilidad (MINBAS).

1.3 Objeto de Estudio.

1.3.1 Descripción General.

En la introducción del documento se formula como **objeto de estudio** para este trabajo de diploma “los procesos de captura de datos de campo del módulo Inventario Nacional de Minerales Sólidos del Sistema de Gestión de Datos Geológicos”. Estos procesos son llevados a cabo hoy día por cada una de las concesiones en el país y dentro de la ONRM.

La elaboración del Inventario Nacional de Minerales Sólidos realizado por la ONRM tiene gran importancia dentro del marco económico, político y social, con el objetivo que la dirección de la Revolución posea toda la información sobre el estado de los recursos minerales existentes en el país. El gobierno cubano, le dedica especial atención a sus recursos y realiza estudios constantemente controlando y garantizando el uso racional de los mismos.

Anualmente los concesionarios envían a la ONRM los datos de su actividad minera realizada, la cual consiste en dos documentos principales, el Informe Inventario y el Informe Geológico, que poseen un formato específico y debido a la complejidad de sus datos hacen que estos informes sean extensos y la posibilidad de introducir errores sea mucho mayor. Estos informes son emitidos en formato duro a la ONRM, donde son revisados manualmente e insertados en un sistema gestor para la confección más tarde del Inventario Nacional de Minerales Sólidos.

1.3.2 Descripción actual del dominio del problema.

La ONRM, como entidad independiente adscrita al Ministerio de la Industria Básica, es la rectora nacional que garantiza la racional explotación y utilización de los recursos minerales. Como parte de la misma se encuentran la Dirección Técnica y el Grupo de Balance, encargados de controlar las actividades y procesos vinculados al Inventario Nacional de Minerales Sólidos.

El Inventario Nacional de Minerales Sólidos se confecciona a partir de los informes entregados por los concesionarios existentes. De acuerdo a la Instrucción del Balance, documento oficial que tiene vigencia desde 1999 (MINBAS, 1999), es una obligación de los concesionarios entregarles al departamento de documentación los datos recogidos en sus concesiones, referentes a la actividad minera realizada (Informe Geológico e Informe Inventario).

El departamento de documentación es el encargado de enviarle a la Dirección Técnica dichos informes que consisten en un grupo de cajas, agrupando los concesionarios por provincias de las cuales, cada una de ellas, posee entre 200 y 400 informes de cada una

de las concesiones, los cuales son encargados de revisar su calidad e integridad. De ser satisfactoria la revisión realizada, se pasa dicho informe al Grupo de Balance para que archiven la información que está comprendida en ellos, de lo contrario, los documentos son regresados al departamento de documentación y éste los entregará al concesionario conjuntamente con las deficiencias que deberá corregir.

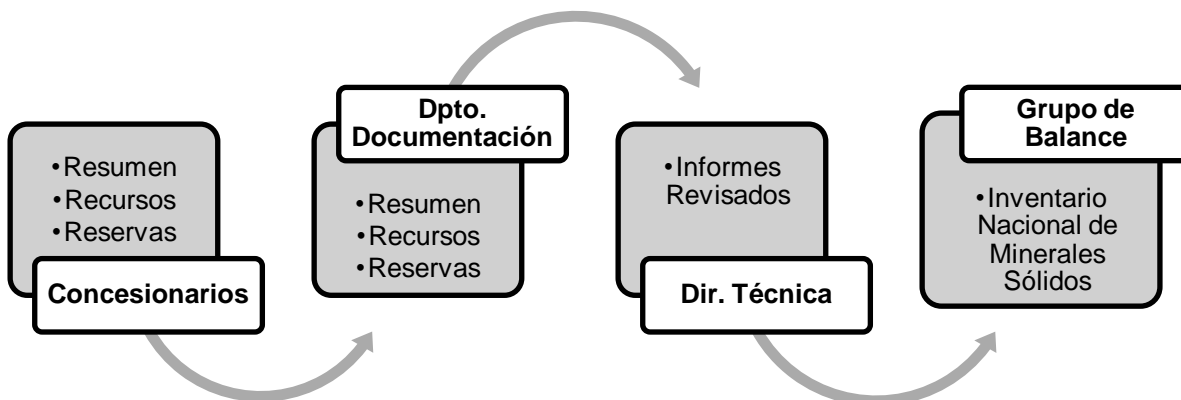


Imagen 1 Curso de los documentos en la ONRM.

Otra de las principales tareas que realiza el Grupo del Balance es la elaboración del Informe Anual del Inventario Nacional de Minerales Sólidos. Este informe está compuesto por la unión de tres informes y se le entrega a la Administración Central del Estado, por lo general en el mes de junio, aunque también puede ser entregado en cualquier etapa del año, si la Administración Central del Estado desea tener una versión más actualizada de los minerales sólidos con que cuenta el país.

El Inventario Nacional de Minerales Sólidos es consultado en varias ocasiones por un amplio grupo de usuarios que pueden ser: trabajadores de la ONRM, concesionarios, empresas extranjeras, instituciones que realizan estudios geológicos y funcionarios del gobierno. La ONRM tiene establecido un conjunto de restricciones para los distintos tipos de usuarios que consultan información del Inventario Nacional de Minerales Sólidos, esto permite un uso racional de la información geológica existente.

La mayoría de los conceptos geológicos con los que se trabaja son establecidos por el gobierno cubano y estipulado en la Gaceta Oficial de la República de Cuba, lo que

concluye que estos datos deben ser siempre capturados y catalogados como está establecido (Asamblea, 1995).

1.3.3 Situación Problemática.

En la ONRM existe actualmente una aplicación que resuelve parcialmente los problemas existentes, la misma no posee las funcionalidades necesarias para la gestión de los datos geológicos, recopilados de cada una de las concesiones existentes en la nación, ni la captura de estos datos por parte de los concesionarios para la confección del Inventario Nacional de Minerales Sólidos. Esta aplicación no cuenta con una interfaz de usuario agradable para los trabajadores del Grupo de Balance.

El sistema gestor de datos actualmente utilizado para la gestión de los datos geológicos es Microsoft Access 97, el cual no permite un número de conexiones concurrentes al gestor de datos por lo que hace que la gestión de los datos (400 informes por provincias aproximadamente) sea realizada solo por una persona a la vez y de manera manual, provocando que este proceso se realice con mayor lentitud. Esta aplicación no cuenta con la funcionalidad de importar un informe en formato digital con la descripción específica de los informes para el Inventario Nacional de Minerales Sólidos, que agilizaría el proceso de gestión de los datos geológicos emitidos por los concesionarios.

1.4 Análisis de otras soluciones existentes.

En la actualidad en la ONRM, el proceso de gestión del Inventario Nacional de Minerales Sólidos cuenta con un sistema gestor de datos en Microsoft Access 97, que le da solución parcialmente al proceso mencionado anteriormente. En el país dentro de las soluciones informáticas desarrolladas para la geología, no existe una aplicación que le dé solución al proceso esbozado anteriormente.

A nivel internacional, las soluciones informáticas desarrolladas en el campo de la geología van encaminadas a sistemas geofísicos y geoespaciales. Se conoce que en Brasil se han automatizado algunos procesos geológicos, pero hasta el momento ninguna para la recopilación de información sobre los minerales existentes. Además se puede añadir que

mundialmente el control de los minerales es realizado por compañías privadas, lo cual trae consigo que los gobiernos de estos países capitalistas no puedan llevar a cabo una eficiente administración de los mismos y por consiguiente se imposibilite el desarrollo de un sistema que gestione esta información.

1.5 Conclusiones parciales.

En este capítulo se presentaron todos los elementos necesarios para tener una mejor comprensión y visión del problema. Se realizó un estudio del proceso de gestión de los datos geológicos y se detallaron los problemas actuales y las condiciones que lo rodean, así como los elementos que afectan este proceso. Se realizó un análisis de la aplicación existente en la actualidad que resuelve parcialmente el problema, así como las deficiencias que este posee.

Añadir además que se especificó completamente la importancia que existe en el tratamiento adecuado de los recursos minerales existentes en la nación, expresado en términos de seguridad y factibilidad económica. También se explicó la necesidad de buscar una solución informática que satisfaga las necesidades actuales para la realización del Inventario Nacional de Minerales Sólidos en la ONRM.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR.

2.1 Introducción.

Desde los inicios de la historia de la informática, con el surgimiento de los gráficos por computadoras, las aplicaciones de escritorio fueron tomando auge y evolucionando hasta nuestros días. En la actualidad las aplicaciones de escritorio funcionan prácticamente de manera independiente, donde muchas de estas aplicaciones bien utilizadas facilitan un gran por ciento del trabajo de los usuarios. Son capaces de manipular datos extremadamente complejos, proporcionar mantenimiento y agilizar relativamente todas las operaciones administrativas y de gestión en una empresa. La ONRM no ha quedado al margen de estos cambios y del desarrollo alcanzado por estas tecnologías.

En el presente capítulo se realizará un análisis de las tecnologías a utilizar para la implementación del software. Se hará énfasis en las conocidas técnicas de software libre y código abierto, el cual ha comenzado desafiando algunas de las más grandes compañías del mundo (Stroube, 2001). Los productos de código abierto y el lucro forman una dupla paradójica (Stroube, 2001). El código abierto implicará la distribución libre y gratuita de programas e información, sin las complicaciones de la gestión de propiedad.

2.2 Software Libre vs. Software privativo.

En un sentido restringido, el software es un conjunto de programas informáticos que al ser ejecutados por una o más computadoras, lleva a cabo el procesamiento, transmisión o almacenamiento de la información. En la actualidad existen grandes diferencias en cuanto al grado de libertad del software, la cual hace posible que surja la siguiente interrogante, ¿qué tipo de software utilizar: libre o privativo? La misma constituye una polémica en la cual debe realizarse un estudio minucioso para saber a ciencia exacta cuál de estas tendencias utilizar. ¿Qué tendencia utilizar para el desarrollo de una aplicación informática en cuanto a criterios de desarrollo y necesidades de los clientes? A continuación, se explican los principales aspectos que conceptualizan a estos tipos de software:

2.2.1 Software privativo.

La expresión “software propietario” proviene del término en inglés “proprietary software.” En la lengua anglosajona, “proprietary” significa poseído o controlado previamente, que destaca la manutención de la reserva de derechos sobre el uso, modificación o redistribución del software. Software privativo, tal como lo ha definido la Free Software Foundation¹ (Fundación para el Software Libre), es cualquier programa que no cumpla con los criterios de la fundación para el software libre. “Privativo”, significa que algún individuo o compañía retiene el derecho de autor (2001) exclusivo sobre una pieza de programación, al mismo tiempo que niega a otras personas el acceso al código fuente del programa y el derecho a copiarlo, modificarlo o estudiarlo.

El término “privativo” alude a que está “poseído y controlado privadamente”. No obstante, el programa puede seguir siendo privativo aunque su código fuente se haya hecho público, si es que se mantienen restricciones sobre su uso, distribución o modificación. Por otro lado, el programa se puede considerar no-privativo, una vez que se haya lanzado con una licencia que permita a otros crear versiones del software modificadas independientemente, sin restricciones onerosas, aún cuando los derechos del autor puedan permanecer en manos de un individuo en particular (2001).

2.2.2 Software libre.

El “software libre” es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

- **Libertad 0:** La libertad de ejecutar el programa, para cualquier propósito.
- **Libertad 1:** La libertad de estudiar cómo trabaja el programa y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.
- **Libertad 2:** La libertad de redistribuir copias para que pueda ayudar al prójimo.
- **Libertad 3:** La libertad de distribuir copias de sus versiones modificadas a terceros. Si lo hace, puede dar a toda la comunidad una oportunidad de

¹ Fundación para el Software Libre.

beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso. También debería tener la libertad de hacer modificaciones y usarlas en privado, en su propio trabajo u obra, sin siquiera mencionar que existen. Si publica sus cambios, no debería estar obligado a notificarlo a alguien en particular, o de alguna forma en particular (gnu.org, 1996).

Al realizar un estudio de las diferentes tendencias de software para el desarrollo de la aplicación, se analizan las características que poseen ambos tipos de software. Se puede mencionar que el software libre es económico en comparación con el software propietario, posee la libertad de uso, redistribución, independencia tecnológica, constituyen ser sistemas más seguros y cuentan con una gran comunidad de desarrollo, la cual presta un soporte continuo y eficiente ante fallos. Todas estas ventajas, que traen consigo el uso del software libre, hace que la balanza se incline hacia éste. Uno de los principales retos en Cuba, es construir software completamente libre, utilizando para ello herramientas libres. A partir de este momento toda la selección que se haga de tecnologías y tendencias será orientada hacia el software libre.

2.3 Lenguajes de programación.

A fin de comprender mejor las ventajas y desventajas de estos lenguajes, es conveniente conocer al menos superficialmente la historia de los mismos, lo cual nos permitirá tener una idea preliminar acerca de su aplicabilidad.

2.3.1 Lenguaje C++.

Bjarne Stroustrup crea una versión experimental denominada "C with Classes" hacia 1979, con la intención de proporcionar una herramienta de desarrollo para el kernel Unix en ambientes distribuidos. En 1983 el lenguaje se rebautiza como C++ y en 1985

Stroustrup publica la primera edición del libro "The C++ Programming Language" (Stroustrup, 1997) que sirvió de estándar informal y texto de referencia. Posteriormente el lenguaje fue estandarizado (ISO C++) y paralelamente son publicadas la segunda y tercera ediciones de "The C++ Programming Language" (Stroustrup, 1997) de modo tal que reflejan estos cambios.

2.3.2 Lenguaje Java.

Java originalmente fue denominado "Oak". Sus inicios datan de 1991 cuando James Gosling (en Sun Microsystems) encabezó un proyecto cuyo objetivo original era implementar una máquina virtual ampliamente portable y un lenguaje de programación, ambos orientados a dispositivos "embedded" (AmericaTI.com, 2006) (procesadores incorporados en diversos dispositivos de consumo masivo como VCR's, tostadoras, PDA's, teléfonos móviles, etc.) La sintaxis del lenguaje heredó características de C++, explícitamente eliminando aquellas que para muchos programadores resultan excesivamente complejas e inseguras (Sun Microsystems, Inc., 2008).

2.3.3 Criterios de Comparación.

Comparar lenguajes de programación nunca ha sido una tarea sencilla u objetiva. Se proporcionan nueve "criterios para la evaluación y comparación de lenguajes (Allen B. Tucker, 1992). Un listado de estos criterios (con ligeras modificaciones) se presenta a continuación:

1. Tipos y estructuras de datos.
2. Modularidad: permitir el desarrollo de componentes independientemente.
3. Portabilidad.
4. Eficiencia.
5. Generalidad: Aplicabilidad, Uso.

2.3.3.1. Tipos y estructuras de datos.

El lenguaje C++ proporciona facilidades que permiten la creación de estructuras de datos muy poderosas y fuertemente integradas en el lenguaje. Por ejemplo, las estructuras contenedoras "clásicas" se proporcionan en su librería de plantillas (STL), asimismo, el desarrollador puede crear sus propios tipos de datos con diversas operaciones asociadas. Gracias a esto, su uso resulta una extensión natural de los tipos de datos primitivos con lo cual se alcanza un alto grado de claridad (Stroustrup, 1997).

Java proporciona tipos de datos primitivos similares (notablemente, careciendo de punteros) y mediante su librería de clases estándar proporciona todas las estructuras contenedoras "clásicas" antes mencionadas, aunque con una sintaxis que pone claramente de manifiesto que se trata de clases auxiliares.

2.3.3.2. Modularidad.

Este criterio estaba referido a la posibilidad de desarrollar componentes de manera independiente los que de manera eventual interactuarían. Ambos lenguajes analizados permiten desarrollar funciones, clases y paquetes de modo independiente, cada cual con sus convenciones particulares. En C++ los conceptos de clase y "espacio de nombres" (namespace) proporcionan dos niveles adicionales de "empacado", mientras que en Java los equivalentes corresponden a las clases y los "paquetes" (Allen B. Tucker, 1992).

2.3.3.3. Portabilidad.

El lenguaje C++ no es un lenguaje automáticamente distribuido en los sistemas Unix, prácticamente todos lo pueden ejecutar ya sea en una variante comercial o mediante el popular GNU GCC/G++ con lo que la disponibilidad está asegurada. En cuanto a su portabilidad, el único inconveniente notorio radica en las implementaciones de la STL. No obstante a lo indicado, ambos lenguajes presentan importantes dificultades de portabilidad, particularmente en cuanto a los siguientes aspectos:

- Características dependientes de la implementación.
- Acceso a librerías del sistema operativo.

En ese sentido Java introdujo un enfoque radical al diseñar un lenguaje prácticamente sin características dependientes del implementador y con una extensa librería utilitaria cuya interfaz de programación está muy fuertemente estandarizada. Esto trajo consigo la famosa promesa: "write once, run everywhere" (escribir una sola vez, ejecutar en cualquier lugar). En conclusión, si es imprescindible una máxima portabilidad a "bajo costo", la respuesta es Java.

2.3.3.4. Eficiencia.

Éste es un aspecto sobre el cual se ha debatido hasta la saciedad y continúa siendo un tema de discusión encarnizada. A continuación se proporcionan algunos argumentos bastante evidentes a favor y en contra de ambos lenguajes. Cuando se refiere a la eficiencia, se hace referencia principalmente a la velocidad con la cual los programas escritos en ambos lenguajes logran llevar a cabo diversas tareas. Asimismo se pueden considerar los recursos del sistema requeridos (principalmente memoria) durante su ejecución.

No es importante ser veloz, sino no ser lento.

Más allá de las referencias y pruebas diversas de "cálculo puro" (en los que Java suele ser más lento que sus contendores) se suele plantear el argumento de la importancia de la velocidad de ejecución del lenguaje en sí. Si bien a todos les interesa que las aplicaciones se ejecuten a máxima velocidad, muchas veces la sensación de velocidad o lentitud no es ocasionada por la performance del "código principal" de la aplicación (que puede estar programado en Java) sino de componentes auxiliares tales como bases de datos, librerías de terceros, dispositivos gráficos acelerados, etc.

En esa línea algunos defensores de Java manifiestan que es poco relevante si el "código C++" es 10 o 50% más veloz, si al final este tiempo no es el verdaderamente percibido por el usuario, asimismo, la aparente reducida eficiencia de Java podría ser frecuentemente superada gracias a la claridad del lenguaje, el cual permitiría implementar mejores algoritmos y de un modo más eficiente.

Claramente, todos estos argumentos son subjetivos (pero muchas veces válidos) y al mismo tiempo son discutibles caso por caso. A modo de conclusión, se considera que un programa en Java suele ser notoriamente más lento si la tarea principal consiste en operaciones lógico/matemáticas, mientras que la performance suele ser ligeramente inferior a la correspondiente a C++ para aplicaciones que hacen uso de muchos otros componentes y librerías auxiliares.

2.3.3.5. Generalidad.

Los dos lenguajes estudiados se proponen como "de propósito general", es decir, serían adecuados para atacar prácticamente cualquier clase de problema. En la práctica, C++ y Java tienen un espectro amplio (por ejemplo, aplicaciones comerciales de toda clase.) notablemente Java, en gran medida gracias a la previsión y publicidad de Sun Microsystem y diversos vendedores de "servidores de aplicación".

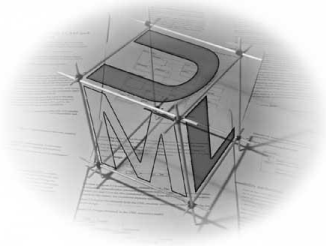
2.4 Fundamentación del lenguaje de programación seleccionado.

Se propone para el desarrollo de la solución el lenguaje Java debido a las características anteriormente plasmadas a las cuales se hace referencia en la siguiente tabla resumen:

Tabla 2 Comparación entre lenguajes de programación.

Características	C++	Java
Tipos y estructuras de datos	Muy Buena	Muy Buena
Modularidad	Muy Buena	Muy Buena
Portabilidad	Buena	Excelente
Eficiencia	Excelente	Buena
Generalidad	Muy Buena	Muy Buena

2.5 El Lenguaje Unificado de Modelado (UML 2.0) como soporte de la modelación de la solución propuesta.



Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar sistemas. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML se aplica en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Desarrollo o RUP), pero no especifica en sí mismo qué metodología o proceso usar. En UML 2.0 hay 13 tipos diferentes de diagramas, para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente. (Imagen 2 Jerarquía de los diagramas UML 2.0.)

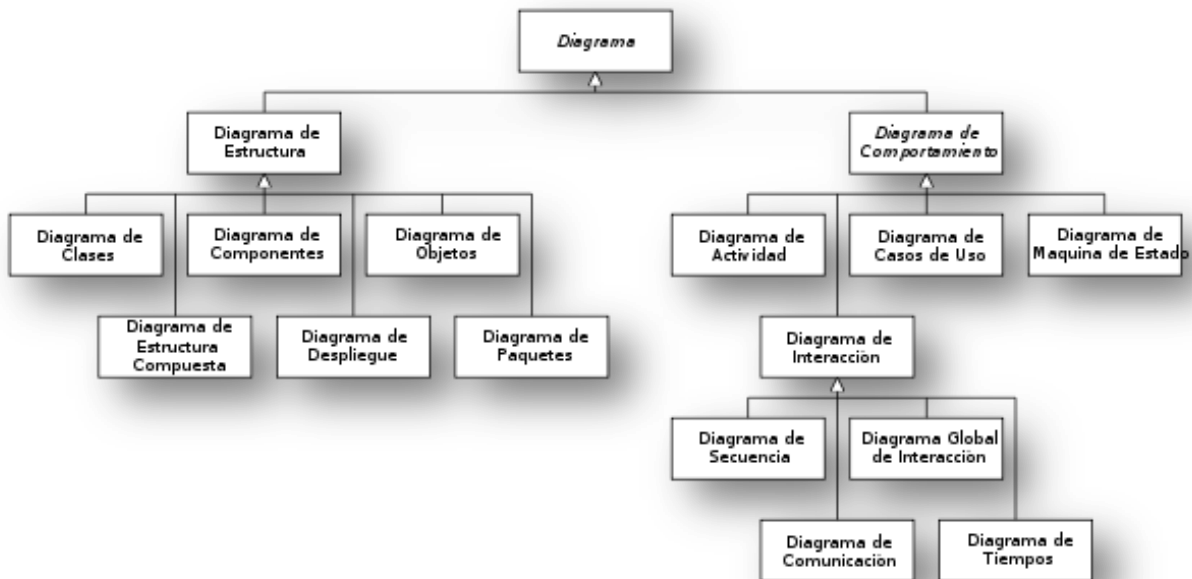


Imagen 2 Jerarquía de los diagramas UML 2.0.

En la actualidad UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Entre más completo es el sistema que se desea crear, más beneficios presenta el uso de UML. El uso de un lenguaje de modelado trae consigo como ventajas fundamentales:

- Diseño y documentación
- Código reutilizable.
- Descubrimiento de fallas.
- Ahorro de tiempo en el desarrollo del software.
- Mucho más fáciles las modificaciones
- Más fácil comunicación entre programadores

2.6 Metodologías para el desarrollo del sistema.

2.6.1 Programación Extrema (XP).

XP² es uno de los llamados procesos o metodologías ágiles de desarrollo de software. Consiste en un conjunto de prácticas llevadas al extremo, con el objetivo de potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. De esta manera los principales objetivos de XP son satisfacer al cliente y potenciar al máximo el trabajo en grupo.

La programación extrema es un enfoque de la ingeniería de software formulado por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software y se basa en la realimentación continua entre el cliente y el equipo de desarrollo y la comunicación fluida entre todos los participantes. La metodología XP se define especialmente para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

² Extreme Programming en idioma español Programación extrema.

2.6.2 RUP.

El Proceso Unificado de Desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental (Jacobson, 2000). (Imagen 3 Los cinco flujos de trabajo – requisitos, análisis, diseño, implementación y prueba- tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición.)

RUP junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto.

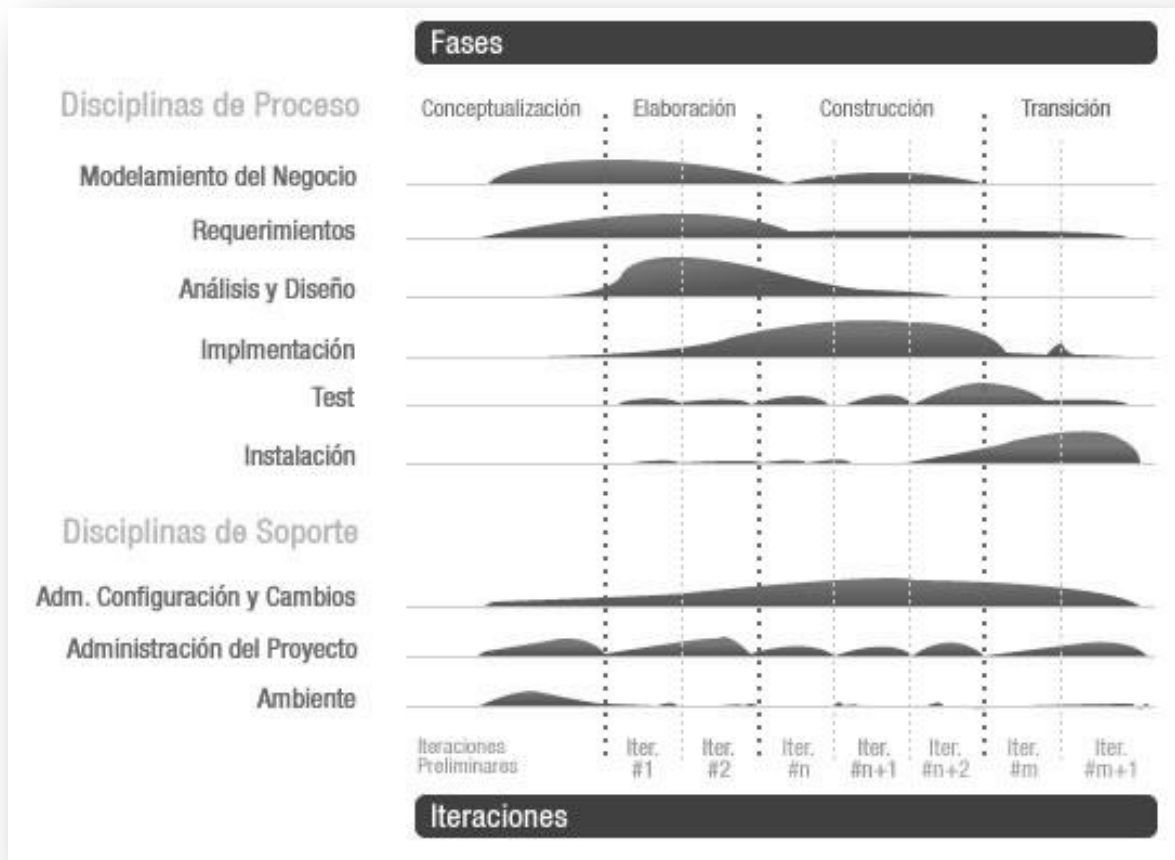


Imagen 3 Los cinco flujos de trabajo – requisitos, análisis, diseño, implementación y prueba- tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición.

Luego de haber analizado profundamente las metodologías más difundidas se decide escoger la metodología RUP como instrumento para el desarrollo de las aplicaciones por parte de la dirección de arquitectura del proyecto. RUP se escoge porque consiste en una metodología que implementa buenas prácticas para el desarrollo de software, permite reducir los riesgos y hace que el proyecto en cuestión se torne más predecible, además de posibilitar la realización de pruebas continuas e iterativas que promueven una mejor evaluación del estado del proyecto.

2.6.3 Fundamentación de la metodología de desarrollo escogida.

RUP es la metodología de desarrollo que más se adapta a la construcción de la aplicación planteada ya que agrupa buenas prácticas de otras metodologías y organiza bien el trabajo orientado a procesos, aspecto que es muy importante ya que se desarrollará describiendo el proceso definido en la introducción de este documento, además, se hace muy necesaria la documentación adecuada de todo el proceso en cada una de sus fases y flujos de trabajo para que se le pueda dar seguimiento en un futuro y de esa forma realizar actualizaciones de la aplicación planteada. De igual manera se documentará y conocerá mucho mejor el negocio descrito en el capítulo anterior.

2.7 Lenguaje XML.



XML es un Lenguaje de Etiquetado Extensible muy simple pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información (W3C, 2008).

2.7.1 ¿Por qué se utilizó XML como fichero para exportar?

Se presentan a continuación algunas características del lenguaje XML, las cuales muestran las facilidades y garantías de su uso, tomándose en cuenta a la hora de elegirlo como formato predeterminado en la elaboración del proyecto:

- XML es un subconjunto simplificado pero estricto de SGML (*Standard Generalized Markup Language*³):
 - Extensible: se pueden definir nuevas etiquetas.

³ Lenguaje de enmarcado estándar generalizado.

- Estructurado: se puede modelar datos a cualquier nivel de complejidad y su definición está en una DTD, *Document Type Definition*⁴.
 - Validable: cada documento se puede validar frente a una DTD, o en su defecto, se puede declarar bien formado.
 - Independiente de medio: para publicar contenidos en múltiples formatos.
 - Independiente de fabricante y de plataforma: para poder utilizar cualquier herramienta estándar.
- XML es fácil de aprender y de usar.
 - Los documentos XML son fácilmente procesables y compartidos en Internet.

2.8 Arquitectura Modelo Vista Controlador (MVC).

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.

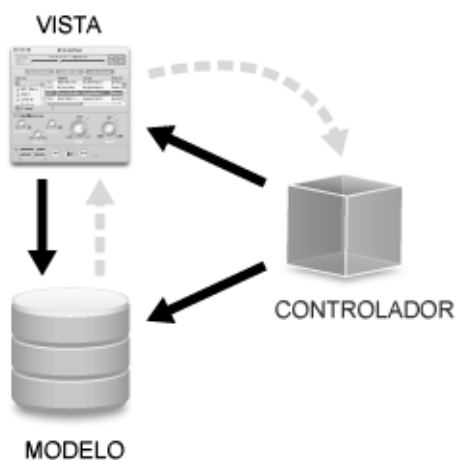


Imagen 4 Arquitectura Modelo Vista Controlador (MVC).

Modelo: Es la representación específica de la información con la cual el sistema opera. El modelo se limita a lo relativo de la vista y su controlador, facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

⁴ Definición del tipo de documento.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Bases de Datos para gestionar los datos: en líneas generales del MVC se corresponde con el modelo. La unión entre capa de presentación y capa de negocio, conocido en el paradigma de la Programación por capas, representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos. MVC no pretende discriminar entre la capa de negocio y presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

2.8.1 ¿Por qué se utilizó como arquitectura MVC?

Para la realización de la aplicación se propone desarrollar el sistema bajo la arquitectura MVC (Imagen 4 Arquitectura Modelo Vista Controlador (MVC).), por su fácil y flexible estructuración del código, “separación de poderes” (datos, implementación e interfaz), seguridad y ayuda para el programador, sencillez para crear distintas representaciones de los datos, facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas, reutilización de los componentes, simplicidad en el mantenimiento de los sistemas, facilidad para desarrollar prototipos rápidos y los desarrollos suelen ser más escalables.

2.9 Propuesta de Herramientas para el desarrollo del sistema.

A continuación se realiza una breve descripción de las herramientas utilizadas para llevar a cabo las diferentes tareas de este proyecto. Para ello, se indica para cada una de ellas, sus funciones y sus principales características.

2.9.1 NetBeans 6.8 M II como IDE para la implementación del sistema.



Un IDE (Entorno de Desarrollo Integrado) 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. NetBeans es un proyecto de código abierto de gran éxito con una gran base de

usuarios, una comunidad en constante crecimiento y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio del 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs⁵ de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándoles nuevos módulos. Debido a que los módulos pueden ser desarrollados de manera independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación.

Entre las características de la plataforma se encuentran:

- Administración de las interfaces de usuario.
- Administración de las configuraciones del usuario.
- Administración del almacenamiento.
- Administración de ventanas.
- Framework basado en asistentes.

2.9.2 Visual Paradigm como herramienta CASE para el modelado del sistema.



Visual Paradigm es una poderosa herramienta CASE que utiliza UML

⁵ Interfaz Programable de la Aplicación.

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

Visual Paradigm presenta como ventajas:

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

2.9.3 HSQLDB como sistema gestor de base de datos.



HSQLDB (Hyperthreaded Structured Query Language Database⁶) es un sistema gestor de bases de datos libre escrito totalmente en Java. Como principal ventaja presenta una gran velocidad de acceso y un reducido tamaño en memoria, este sistema gestor de datos puede mantener la base de datos en memoria o en ficheros en disco. La suite ofimática OpenOffice.org lo incluye desde su versión 2.0 para dar soporte a la aplicación Base, equivalente de Access dentro de Open Office.

HSQLDB presenta como ventajas:

- Escrito por completo en Java.

⁶ Lenguaje de consultas altamente rápido.

- Completo sistema gestor de bases de datos relacional.
- Tiempo de arranque mínimo y gran velocidad en las operaciones: SELECT, INSERT, DELETE y UPDATE.
- Sintaxis SQL estándar.
- Integridad referencial (claves foráneas).
- Procedimientos almacenados en Java.
- Disparadores (Triggers).
- Tablas en disco de hasta 8GB.

2.10 Sistema operativo para el desarrollo de la aplicación.



Un sistema operativo (SO) actúa como interfaz entre un usuario de una computadora y el hardware de la misma; permitiendo la posibilidad al usuario de la ejecución de programas. Por tanto, se puede establecer como objetivo fundamental de un SO, la optimización de todos los recursos del sistema, lo que incluye tanto hardware como software. En diferentes ocasiones se encuentra como un administrador de los recursos de hardware y el encargado de presentar al usuario una capa de abstracción de software o máquina virtual que se encargue de aislar a los usuarios de la complejidad del hardware.

Existen numerosos sistemas operativos libres reconocidos a nivel mundial con los que los usuarios pueden realizar innumerables acciones en su beneficio, para citar algunos tenemos: Linux, Unix, FreeBSD entre otros. Hoy en día los sistemas operativos libres más usados son aquellos desarrollados sobre la plataforma GNU/Linux, cuentan con varias distribuciones como Debian, Gentoo, Suse, Red Hat entre otras. De todas, la más estable y óptima para el desarrollo de aplicaciones informáticas es el Debian.

Ubuntu es una distribución Linux que ofrece un sistema operativo enfocado a computadoras de escritorio, aunque también proporciona soporte para servidores, está basado en Debian GNU/Linux y concentra su objetivo en hacer de Linux un sistema operativo más accesible y fácil de usar. Por lo antes mencionado se selecciona como

sistema operativo al Ubuntu en su versión 9.10 (Karmic Koala) para el desarrollo de la aplicación que dará solución al problema planteado.

2.11 Framework para el desarrollo del sistema.

De forma general, un framework para el desarrollo de una aplicación, es un conjunto de clases que cooperan y forman un diseño reutilizable, formando una infraestructura que facilita y agiliza el desarrollo de aplicaciones.

2.11.1 Swing.

Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegados y tablas. El paquete Swing es parte de la JFC (Java Foundation Classes⁷) en la plataforma Java. La JFC facilita la construcción de las interfaces (GUIs). Los componentes Swing se identifican porque pertenecen al paquete javax.swing (2009).

Swing existe desde la JDK 1.1. Antes de la existencia de Swing, las interfaces gráficas con el usuario se realizaban a través de AWT (Abstract Window Toolkit⁸), de quien Swing hereda todo el manejo de eventos. Usualmente, para toda componente AWT existe una componente Swing que la reemplaza, por ejemplo, la clase Button de AWT es reemplazada por la clase JButton de Swing (el nombre de todas las componentes Swing comienza con "J"). Los componentes de Swing utilizan la infraestructura de AWT, incluyendo el modelo de eventos AWT, el cual rige cómo un componente reacciona a eventos (2009).

Swing presenta como ventajas:

- 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.

⁷ Fundación de Clases para Java.

⁸ Librería gráfica Java.

2.12 Conclusiones parciales.

Para brindarle solución a los problemas planteados urge la necesidad de implementar un sistema informático que permita de manera fiable la captura de los datos de campo para la realización del inventario de minerales sólidos. En este capítulo se realizó un completo análisis de las principales tendencias y tecnologías necesarias para llevar a cabo la implementación del sistema, siempre teniendo en cuenta las necesidades del cliente, las preferencias actuales para desarrollar aplicaciones de escritorio y las posibles herramientas y soluciones mediante la utilización del software libre.

Por lo antes explicado se decide desarrollar el sistema sobre el sistema operativo Ubuntu en su versión 9.10, utilizando como plataforma Java, como IDE para la implementación NetBeans 6.8, como sistema gestor de datos HSQLDB, siguiendo como metodología RUP y utilizando como herramienta para la modelación del sistema Visual Paradigm. Los ficheros serán generados utilizando el formato universal XML.

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN.

3.1 Introducción.

Para el desarrollo exitoso del sistema es necesario tener un amplio conocimiento de los procesos que ocurren en la actualidad en la captura de datos de campo en los concesionarios, así como identificar los principales requisitos funcionales y no funcionales que debe tener el sistema para posteriormente detallar sus principales procesos. Aspectos como los antes mencionados se comprenden mejor mediante la modelación, utilizando un conjunto de normativas y artefactos que proporciona el RUP.

En el presente capítulo se realizará el modelado del negocio referente al sistema, se identificarán y establecerán el conjunto de requisitos funcionales y no funcionales existentes, necesarios para el desarrollo del sistema y se modelará el sistema completamente, todos estos con sus artefactos correspondientes.

3.2 Modelo de Negocio.

El Modelo de Negocio está formado por actores, casos de uso y relaciones entre ambos, describe lo que el sistema debería hacer por sus usuarios y bajo qué restricciones (Jacobson, 2000). El modelo del negocio permite comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema, además de los problemas actuales de la organización, e identificar así las mejoras potenciales. A continuación se introducen los actores y trabajadores del negocio con sus principales características.

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. Lo que se modela como actor del negocio es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (Jacobson, 2000). A continuación se muestra una tabla con los actores identificados en el presente negocio:

Tabla 3 Actores del Negocio y su Descripción.

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

Concesionario	Concesionario o Titular es un Cliente Externo o Entidad Geológica que ya tiene otorgada una concesión.
---------------	--

Un *trabajador* representa un puesto que puede ser asignado a una persona o equipo y que se requiere responsabilidades y habilidades como realizar determinadas actividades o desarrollar determinados artefactos (Jacobson, 2000). A continuación se muestran los trabajadores identificados en el negocio:

Tabla 4 Trabajadores del Negocio y su Descripción.

Trabajador	Descripción
Concesionario	Concesionario o Titular es un Cliente Externo o Entidad Geológica que ya tiene otorgada una concesión.
Departamento de Documentación	Es el departamento encargado de recibir y despachar todos los documentos en la ONRM.
Dirección Técnica	Es el departamento encargado en la ONRM de revisar y aprobar los Informes Geológicos que entregan los Concesionarios en el departamento de documentación dentro de dicha oficina.
Grupo de Balance	Es el encargado de actualizar la información que se utiliza para la elaboración del Inventario Nacional de Minerales Sólidos y de confeccionar el mismo.

3.2.2 Procesos del negocio.

En el negocio se identifican dos procesos, los mismos están conformados cada uno por un caso de uso del negocio que lo identifica y describe, a continuación se muestran estos casos de uso con sus descripciones:

1) *Proceso de Captura de Datos de Campo por parte de los concesionarios:*

Caso de Uso del Negocio Registrar Datos de Campo: Un concesionario cumpliendo con sus obligaciones debe entregar un grupo de informes geológicos a la ONRM, para el cual debe registrar los datos de toda la actividad minera realizada en sus concesiones durante un período de tiempo el cual consiste en dos informes, Informe Inventario e Informe Geológico (MINBAS, 1999).

2) *Proceso de Entrega de Informes Geológicos por parte de los concesionarios:*

Caso de Uso del Negocio Entregar Informes Geológicos: El concesionario, cumpliendo con sus obligaciones según la Instrucción del Balance (MINBAS, 1999) entrega a la ONRM un informe de toda la actividad minero-geológica realizada en sus concesiones (Informe Geológico e Informe Inventario).

3.2.3 Diagrama de Casos de Uso del Negocio.

A continuación se presenta en (Imagen 5 Diagrama de Casos de Uso del Negocio.) el diagrama de casos de uso del negocio, el cual está constituido por el actor del negocio y los casos de uso que representan los principales procesos identificados anteriormente.

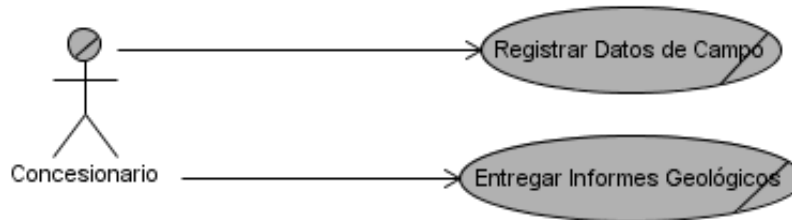


Imagen 5 Diagrama de Casos de Uso del Negocio.

3.2.4 Descripción textual de los Casos de Uso del Negocio.

Tabla 5 Descripción textual del Caso de Uso Registrar Datos de Campo.

Caso de Uso:	Registrar Datos de Campo
Actores:	Concesionario
Trabajadores:	Concesionario
Resumen:	Un concesionario cumpliendo con sus obligaciones debe entregar un grupo de informes geológicos a la ONRM, para el cual debe registrar los datos de toda la actividad minera realizada en sus concesiones durante un período de tiempo el cual consiste en dos informes, Informe Geológico e Informe Inventario (MINBAS, 1999).
Precondiciones:	-

Flujo Normal de los Eventos			
Actor		Negocio	
1	El Concesionario registra los datos de campo.	1.1	El Concesionario registra los datos de toda la actividad minera geológica realizada.
		1.2	Prepara el Informe Geológico e Informe Inventario.
Flujo Alterno de los Eventos			

Tabla 6 Descripción textual del Caso de Uso Entregar Informes Geológicos.

Caso de Uso:	Entregar Informes Geológicos		
Actores:	Concesionario		
Trabajadores:	Departamento de Documentación, Dirección Técnica, Grupo de Balance		
Resumen:	El concesionario cumpliendo con sus obligaciones según la Instrucción del Balance (MINBAS, 1999) entrega a la ONRM dos informes con toda la actividad minero-geológica realizada en sus concesiones (Informe Geológico e Informe Inventario).		
Precondiciones:	-		
Flujo Normal de los Eventos			
Actor		Negocio	
1	El Concesionario entrega los Informes Geológicos al Departamento de Documentación.	1.1	El Departamento de Documentación recibe los Informes Geológicos.
		1.2	El Departamento de Documentación envía los Informes Geológicos a la Dirección Técnica.

		1.3	La Dirección Técnica revisa la integridad de los datos de los Informes recibidos.
		1.4	De ser satisfactoria la revisión realizada se aprueban los Informes Geológicos y se entregan al Grupo de Balance.
		1.5	El Grupo de Balance archiva la información comprendida en los Informes.
Flujo Alternativo de los Eventos			
		1.3	En caso que la revisión de los Informes Geológicos no sea satisfactoria son devueltos al Departamento de Documentación con las deficiencias que debe corregir el concesionario.
		1.4	El Departamento de Documentación entrega los Informes Geológicos al concesionario con las deficiencias que debe corregir.

3.2.5 Diagrama de Objetos del Negocio.

Se muestra el diagrama de objeto del negocio (Imagen 6 Diagrama de Objetos del Negocio.) donde se observan claramente los trabajadores en su interacción con los principales objetos que se han identificado.

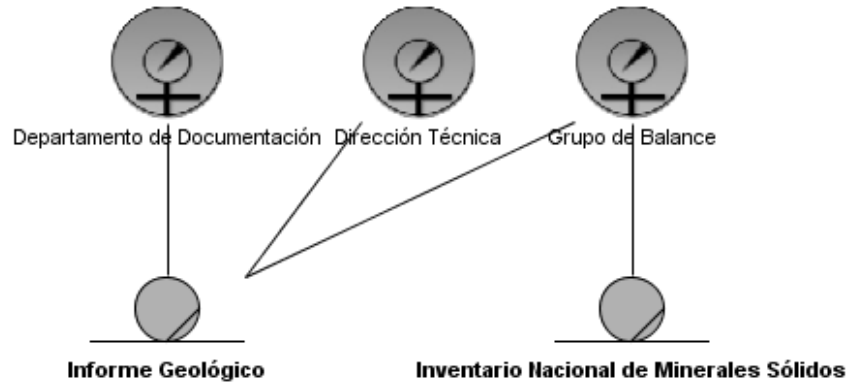


Imagen 6 Diagrama de Objetos del Negocio.

3.3 Requisitos funcionales y no funcionales.

En el siguiente acápite se muestran en forma de listado los distintos requisitos funcionales y no funcionales del sistema.

3.3.1 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. En la realización de los casos de uso del negocio se obtienen actividades que serán objeto de automatización, las cuales son el punto de partida para identificar lo que debe hacer el sistema.

R.1. Gestionar Recursos y Reservas.

R.1.1. Adicionar Recursos y Reservas.

R.2. Gestionar Informe Geológico.

R.3. Importar Informe Digital.

R.3.1. Actualizar Datos Geológicos.

R.3.2. Comprobar Datos.

R.3.3. Mostrar Resultado de la Importación.

R.4. Importar Configuración.

R.5. Exportar Informe Digital.

R.6. Generar Informe Geológico.

R.7. Generar Informe de Recursos y Reservas.

R.8. Imprimir Informes Geológicos.

3.3.2 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 (Jacobson, 2000). Puede especificar restricciones físicas sobre un requisito funcional. A continuación se listan los referentes al sistema a desarrollar.

Apariencia o interfaz externa.

- El diseño de la interfaz deberá realizarse de forma sencilla y permitirles a todas las personas, incluso sin mucha experiencia en el manejo de computadoras, su fácil utilización.
- Los estilos y colores a utilizar serán los representativos de la ONRM.

Usabilidad.

- El sistema será usado por cualquier persona que tenga un conocimiento mínimo de computación.
- El sistema podrá captar información de informes digitales para que el usuario final no tenga la necesidad de teclear toda la información de los informes a través de formularios.

Portabilidad.

El sistema será multiplataforma.

Rendimiento.

- Tanto el tiempo de respuesta como la velocidad de procesamiento de la información deberán ser rápidos, menos de 30 segundos.
- El sistema requerirá de un buen rendimiento en máquinas de pocos recursos de Hardware (1 GHz de microprocesador y 256 MB de memoria RAM).

Soporte.

El sistema deberá contar con un manual de usuario.

Confiabilidad y Disponibilidad.

El sistema deberá estar disponible durante todo el horario de trabajo que tengan los usuarios finales.

Software.

La PC⁹ cliente tendrá sistemas operativos de las plataformas Windows y Linux, cualquiera de sus versiones. Se recomienda la instalación del entorno virtual para java en su versión 1.6 o superior.

Hardware.

La PC cliente deberá contar con un microprocesador con una velocidad de procesamiento superior a un 1 GHz¹⁰ y una memoria RAM¹¹ de 256 MB o superior.

3.4 Descripción del sistema propuesto.

La propuesta de solución es un sistema que permita la gestión de los datos geológicos. Este sistema será desplegado en los concesionarios y la ONRM y consiste en una aplicación de escritorio, la cual permitirá la gestión de la actividad minera realizada por los concesionarios. Mediante este sistema la Dirección Técnica en la ONRM podrá revisar de manera eficiente los informes emitidos por los concesionarios y los concesionarios podrán contar con una aplicación para la gestión de la actividad minera realizada y, de esta manera, generar los Informes Geológicos que deben entregar según la Instrucción del Balance (MINBAS, 1999).

3.4.1 Descripción de los actores.

Un actor es la idealización de una persona externa, un proceso o cosa que interactúa con un sistema, subsistema, o clase. Caracteriza a las interacciones que pueden tener los usuarios con el sistema. Se puede incluir que un usuario puede llegar a desempeñarse como varios actores en tiempo de ejecución dentro del sistema, además que diferentes

⁹ Personal Computer: en idioma español, Computadora Personal.

¹⁰ Gigahercio: es un múltiplo de la unidad de medida de frecuencia hercio (Hz) y equivale a 109 Hz.

¹¹ Memoria de Acceso Aleatorio.

usuarios pueden estar reflejados en un mismo actor y por lo tanto, representan distintas instancias de la misma definición de actor. Cada actor participa en uno o más casos de uso. Se muestran a continuación en la (Tabla 7 Actores del Sistema y su Descripción.) los actores del sistema con su respectiva descripción.

Tabla 7 Actores del Sistema y su Descripción.

Actor	Descripción
Concesionario	Es el encargado de registrar los datos de la actividad minera en el concesionario como entidad.
Administrador de Balance	Es el encargado de revisar y archivar los Informes Geológicos para la confección del Inventario Nacional de Minerales Sólidos.

3.4.2 Casos de uso del sistema.

A continuación se muestra el Diagrama de Caso de Uso del Sistema (Imagen 7 Diagrama de Casos de Uso del Sistema.), en el mismo están representados los distintos actores que interactúan con el sistema. También se puede encontrar un conjunto de paquetes que engloban todos los Casos de Uso del Sistema (representan los principales procesos del mismo) los cuales constituyen una funcionalidad dentro de dicho sistema a desarrollar. En los siguientes acápite se muestra el contenido de cada paquete, así como la descripción textual de cada caso de uso del mismo.

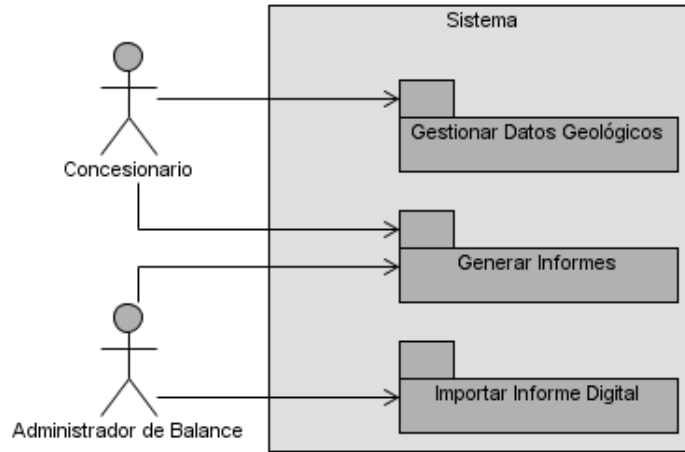


Imagen 7 Diagrama de Casos de Uso del Sistema.

3.4.2.1. Paquete Gestionar Datos Geológicos.

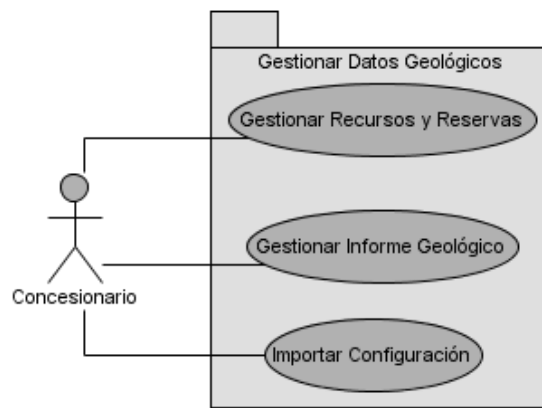


Imagen 8 Diagrama de Casos de Usos del Sistema paquete Gestionar Datos Geológicos.

Descripción textual de los Casos de Uso.

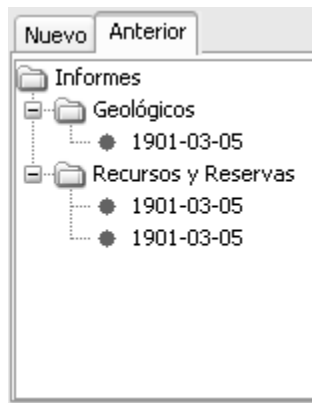
Tabla 8 Descripción textual del Caso del Uso Gestionar Recursos y Reservas.

Caso de Uso: Gestionar Recursos y Reservas	
Actores:	Concesionario.
Resumen:	El caso de uso se inicia cuando el Concesionario inicia la aplicación, la misma le muestra al usuario los campos necesarios para la gestión de los recursos y las reservas y los datos de los

	Depósitos, Sectores y Unidades contenidas en su Concesión, así como sus materias primas, usos y parámetros de calidad asignados.
Precondiciones:	El Concesionario debe poseer la configuración del sistema definida para su Concesión.
Referencias:	R.1. R.1.1. R.1.2. R.1.3. R.4
Prioridad:	Crítico.

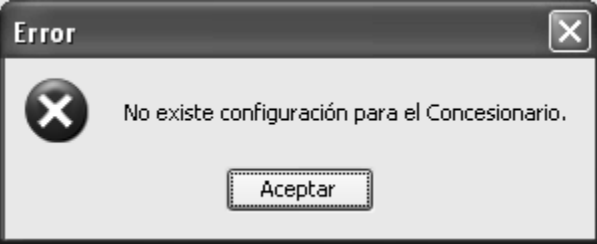
Flujo Normal de los Eventos

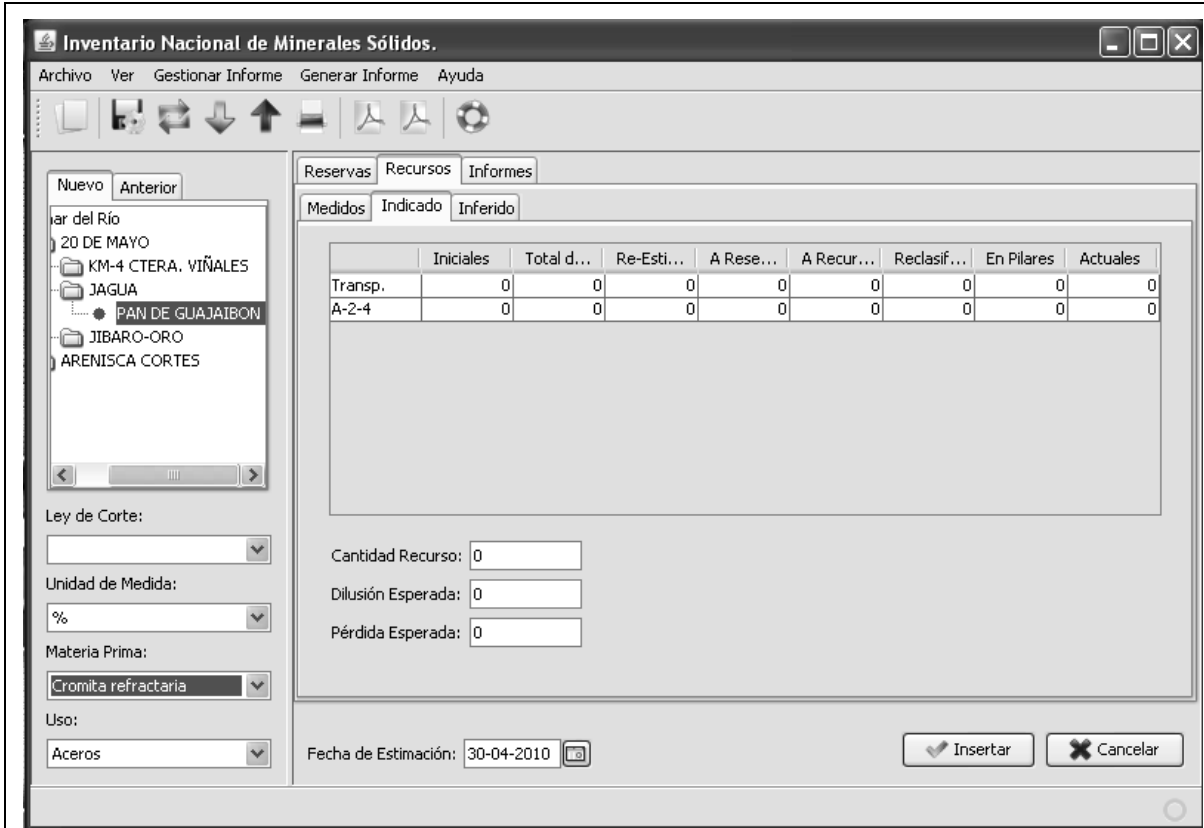
Actor		Sistema	
1	El Concesionario inicia la aplicación accediendo al ejecutable o accesos directos.	1.1	El sistema verifica si posee la configuración predeterminada para el concesionario.
		1.2	El sistema muestra los depósitos en posesión del concesionario y presta la opción de seleccionar el sector, la unidad, la materia prima, los usos y sus parámetros de calidad asignados.
		1.3	El sistema muestra un árbol con los informes gestionados con anterioridad.



Flujo Alternativo de los Eventos

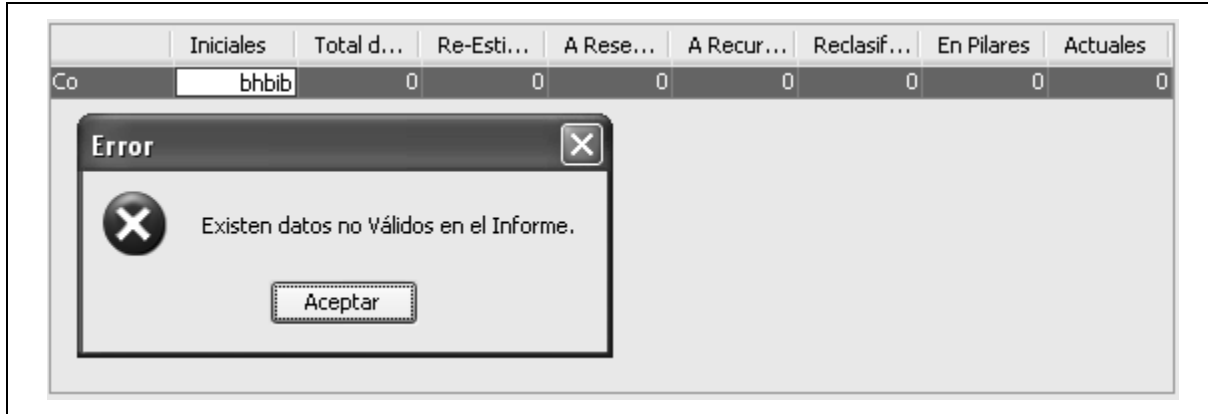
Actor		Sistema	
		1.2	El sistema Informa que no posee la

			configuración predeterminada para el Concesionario.
			
Adicionar Recursos y Reservas			
Actor		Sistema	
2	El Concesionario accede a Gestionar Recursos y Reservas.	2.1	El sistema muestra un formulario con los campos necesarios.
3	El Concesionario llena los campos.		
4	El Concesionario presiona el botón Adicionar para insertar los datos.	4.1	El sistema valida los campos.
		4.2	El sistema adiciona los datos de recursos y reservas.



Flujo Alternativo de los Eventos

Actor		Sistema	
		4.1	El sistema valida los campos.
		4.2	El sistema Informa la existencia de errores en los campos a insertar.
4	El Concesionario presiona el botón Cancelar para cancelar la gestión de los datos geológicos.	4.1	El sistema limpia los campos del formulario.



3.4.2.2. Paquete Generar Informes.

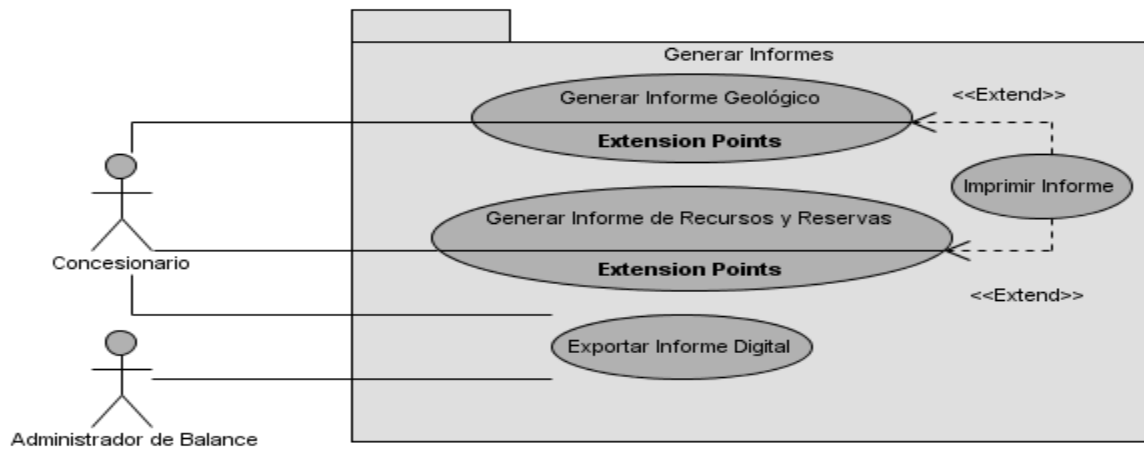


Imagen 9 Diagrama de Casos de Uso del Sistema paquete Generar Informes.

Descripción textual de los Casos de Uso.

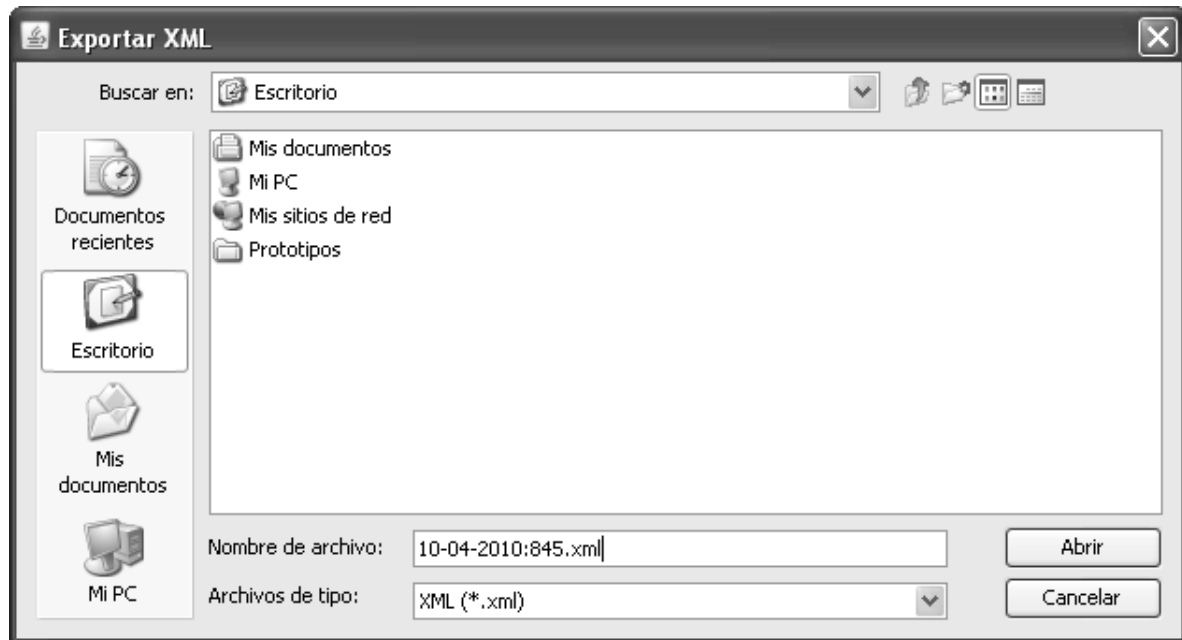
Tabla 9 Descripción textual del Caso de Uso Exportar Informe Digital.

Caso de Uso: Exportar Informe Digital	
Actores:	Concesionario.
Resumen:	El Caso de Uso se inicia cuando el Concesionario accede al submenú Exportar XML, del menú Archivo.
Precondiciones:	El sistema debe tener los datos necesarios para el informe digital que desea ser generado.

Referencias:	R.5.
Prioridad:	Crítico

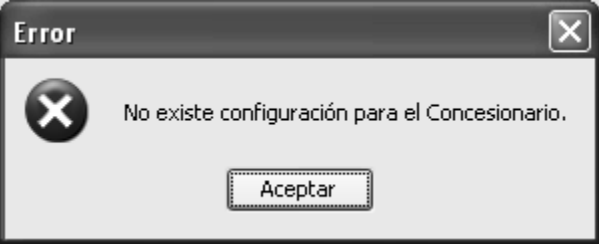
Flujo Normal de los Eventos

Actor		Sistema	
1	El Concesionario accede al submenú Exportar XML, del menú Archivo.	1.1	El sistema verifica la existencia de los datos geológicos necesarios.
		1.2	El sistema muestra un formulario para generar el informe digital con las opciones Aceptar y Cancelar.
2	El Concesionario provee la dirección donde desea guardar el fichero XML y presiona Aceptar.	2.1	El sistema procesa los datos geológicos correspondientes y genera el Informe Digital en formato XML.



Flujo Alternativo de los Eventos

Actor		Sistema	
		1.2	El sistema muestra un cuadro de diálogo informando al Concesionario que no existen

			datos en la base de datos para generar el XML.
2	El concesionario presiona el botón Cancelar.	2.1	El sistema cierra la ventana para el informe digital y mantiene los datos anteriores.
			

3.4.2.3. Paquete Importar Informe Digital.

3.5 Conclusiones parciales.

Se presentó durante el capítulo toda la información perteneciente al negocio, se describieron los procesos con la especificación de sus datos, actividades y personas involucradas, así como los casos de uso del negocio. Se realizó además el modelo del sistema, comenzando por los requisitos funcionales y los no funcionales, que constituyen el paso fundamental para adentrarse en este modelo. Se presentó el Diagrama de Casos de Uso del Sistema por paquetes, cada uno de los paquetes por casos de uso y la respectiva descripción de cada uno de los casos de uso, haciendo evidente la forma en la que queda estructurado el sistema a desarrollar.

CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

4.1 Introducción.

En el presente capítulo se adentrará en el desarrollo del sistema propuesto anteriormente. Para ello se construirá el modelo de diseño, el cual constituye un plano cercano a la implementación del sistema y favorecerá a una arquitectura estable y sólida. Como parte del modelo de diseño serán realizados los diagramas de clases del diseño, basados en la implementación utilizada por el lenguaje Java para la arquitectura MVC. Los diagramas de clases del diseño serán la base para los diagramas de componentes y el diagrama de despliegue, los cuales serán presentados.

Con el desarrollo del siguiente capítulo quedarán definidos los principios de diseño, así como los estándares de la interfaz de la aplicación y se presentará el diseño de la base de datos que constituye el medio de almacenamiento temporal de la información a generar en el fichero XML. Luego de realizar los diagramas necesarios para la implementación del sistema se desarrollarán las técnicas que serán utilizadas para realizar pruebas al sistema.

4.2 Modelo de diseño.

El modelo de diseño es planteado como un modelo de objetos que describe la realización física de los casos de uso, centrándose en los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, que tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación (Jacobson, 2000).

4.2.1 Diagramas de diseño.

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y las relaciones entre ellas. Los diagramas de clases son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido. Cuando se crea un diagrama de clases,

se está modelando una parte de los elementos y relaciones que configuran la vista de diseño del sistema.

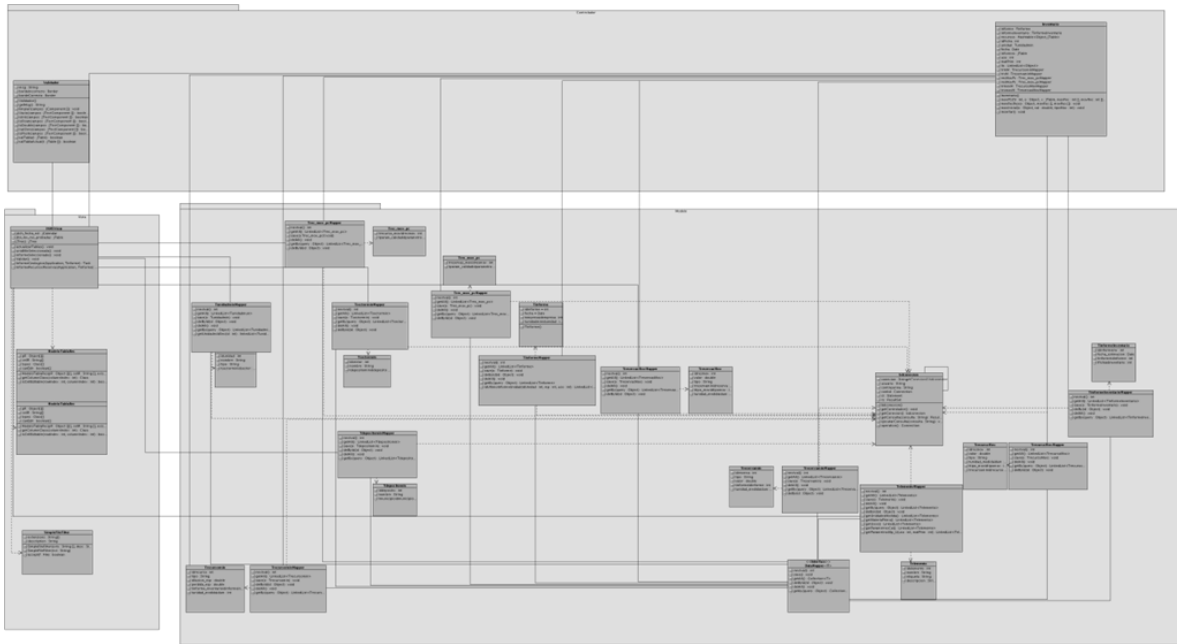


Imagen 10 Diagrama de Clases del Diseño del Caso de Uso Gestionar Informe Geológico.

4.3 Principios de diseño.

Para desarrollar una herramienta atractiva para el usuario es necesario definir algunos principios de diseño, los cuales no solo definen la apariencia estética de la interfaz, sino que posibilitan también que la misma brinde adecuadamente información que le sea útil al usuario, para ello, es fundamental un contenido bien distribuido. Un buen diseño gráfico, una buena navegabilidad y una adecuada usabilidad traen consigo una interfaz bien diseñada. A continuación se definen los principios de diseño a tener en cuenta en la construcción de la herramienta propuesta.

- 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.

- 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.
- Los 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.

4.3.1 Estándares de la interfaz de la aplicación.

La interfaz de una aplicación, más allá del diseño de componentes visuales, tiene como principal objetivo facilitar la interacción entre el usuario y la aplicación. Una correcta distribución del contenido, facilita la manipulación y acceso a sus diferentes funcionalidades. Con el objetivo de garantizar un mejor diseño y una adecuada usabilidad, se definen una serie de estándares que deben ser aplicados en el diseño de la interfaz. Teniendo en cuenta que la herramienta que se desarrolla es una aplicación de escritorio y que los usuarios finales tienen conocimientos básicos de computación, la interfaz de la misma debe cumplir con los estándares que se listan a continuación:

- Barra de Título.
- El título a la izquierda y los botones minimizar, maximizar y cerrar, ubicados del lado derecho.
- No deben usarse colores degradados u otros que interrumpen la visibilidad de los elementos anteriores.
- Barra de Menú.
- Los menús en esta barra se ordenan según la importancia de las funcionalidades que brindan.
- El primer menú es "Archivo" y en él la última opción es "Salir".

- Tiene un menú Ayuda con dos elementos: “Temas de ayuda” y un cuadro de diálogo "Acerca de".
- Incluye atajos para los elementos del menú más frecuentemente usados.
- Para no cargar con demasiados elementos el menú, se usan submenús o líneas divisorias para lograr un acceso rápido.
- Barra de Herramientas.
- Se usarán iconos de tamaño medio con etiquetas que brindan una mejor comprensión a los usuarios.
- Gráficos simples y fáciles de entender.
- Barra de Estado.
- Se muestra el estado actual de las acciones que realiza el usuario.
- Punteros del Ratón.
- Se usan los punteros estándares del sistema operativo en el cual se esté ejecutando la aplicación.

4.3.2 Concepción general de la ayuda.

El software a implementar contará con una ayuda general para facilitar la manipulación de la herramienta por parte de los usuarios finales. La misma se encontrará situada en el menú principal para lograr una localización rápida, la estructura que se propone para organizar su contenido es la siguiente:

1. 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 por sus siglas en inglés):

2. 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 es posible lograr que de los objetos mantenga su valor en el tiempo y el espacio, las clases identificadas durante el diseño no necesariamente tienen que ser persistentes, durante el diseño se identifican 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 el diagrama de clases persistentes referentes a la aplicación a desarrollar y posteriormente se representa el diagrama entidad relación de la base de datos generado por el diagrama anterior.

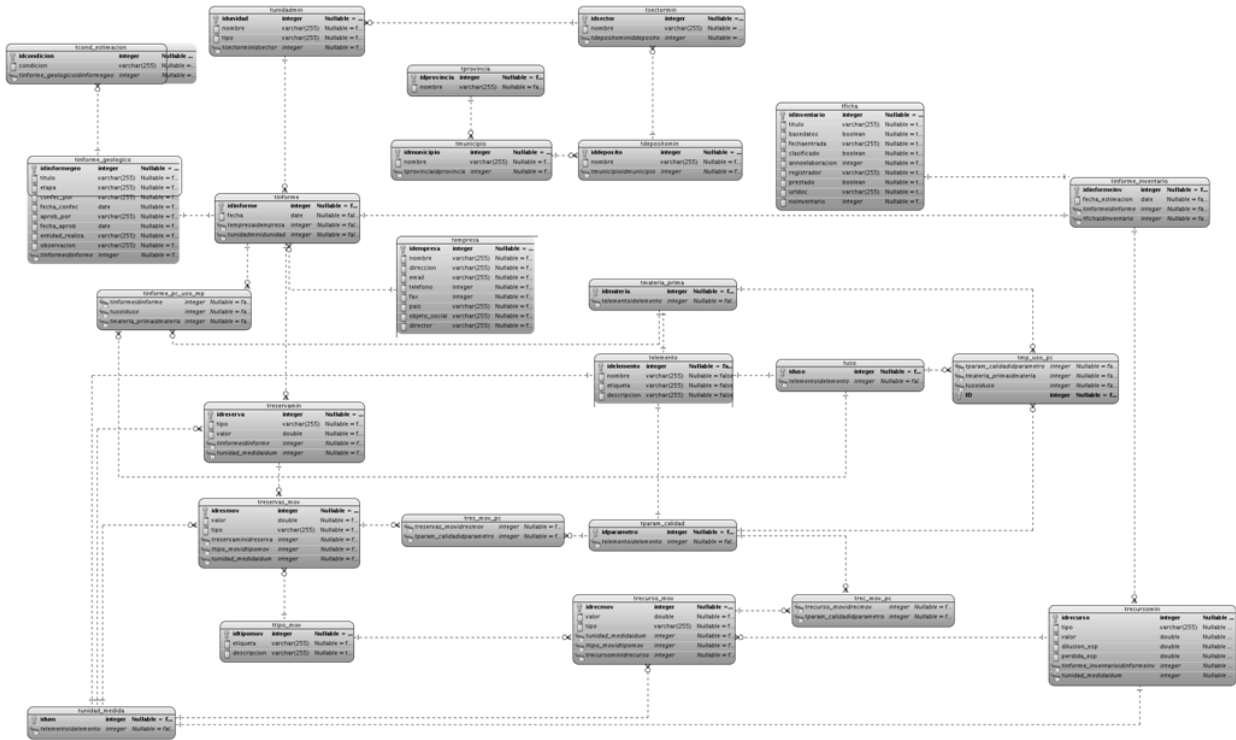


Imagen 11 Diagrama Entidad - Relación.

4.5 Modelo de despliegue.

El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. El modelo consiste en uno o más nodos, dispositivos y conectores entre estos. El modelo de despliegue también mapea procesos dentro de estos elementos de procesamiento, permitiendo la distribución del comportamiento a través de los nodos que son representados. A continuación se muestra el diagrama de despliegue modelado para la aplicación a desarrollar.

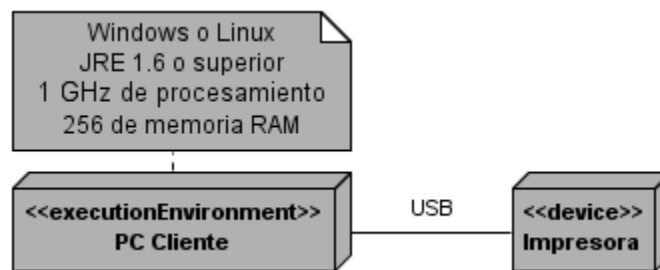


Imagen 12 Diagrama de Despliegue.

4.6 Modelo de Implementación.

El modelo de implementación está constituido por una colección de componentes, subsistemas de implementación, así como paquetes utilizados para agrupar elementos del modelo. Los componentes son la parte modular del sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos, estos pueden ser un fichero de código fuente, ficheros de código binario, ejecutables y similares. Los diagramas de componentes son utilizados para modelar la vista estática del sistema, mostrando la organización y las dependencias lógicas entre los componentes.

El modelo de datos para cada una de las tablas de la base de datos fueron construidos basados en el modelo genérico (Imagen 13 Diagrama de Componentes Modelo Genérico.), para cada una de las tablas de la base de datos existen dos ficheros físicos

como se muestra a continuación. Estos diagramas han sido confeccionados por separado, uno por cada caso de uso. A continuación se muestran el diagrama correspondiente al caso de uso (Imagen 14 Diagrama de Componentes del Caso de Uso Generar Informe Recursos y Reservas.).

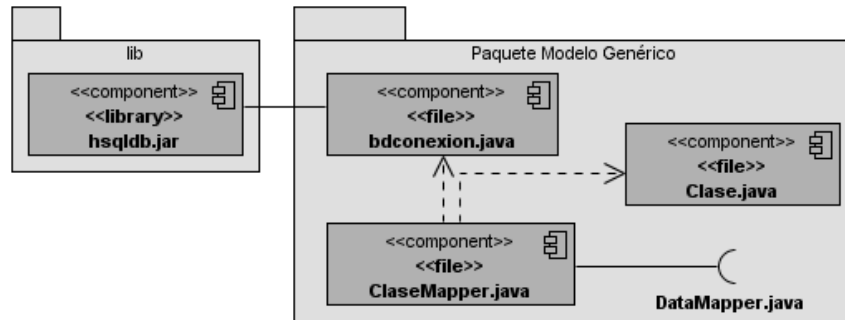


Imagen 13 Diagrama de Componentes Modelo Genérico.

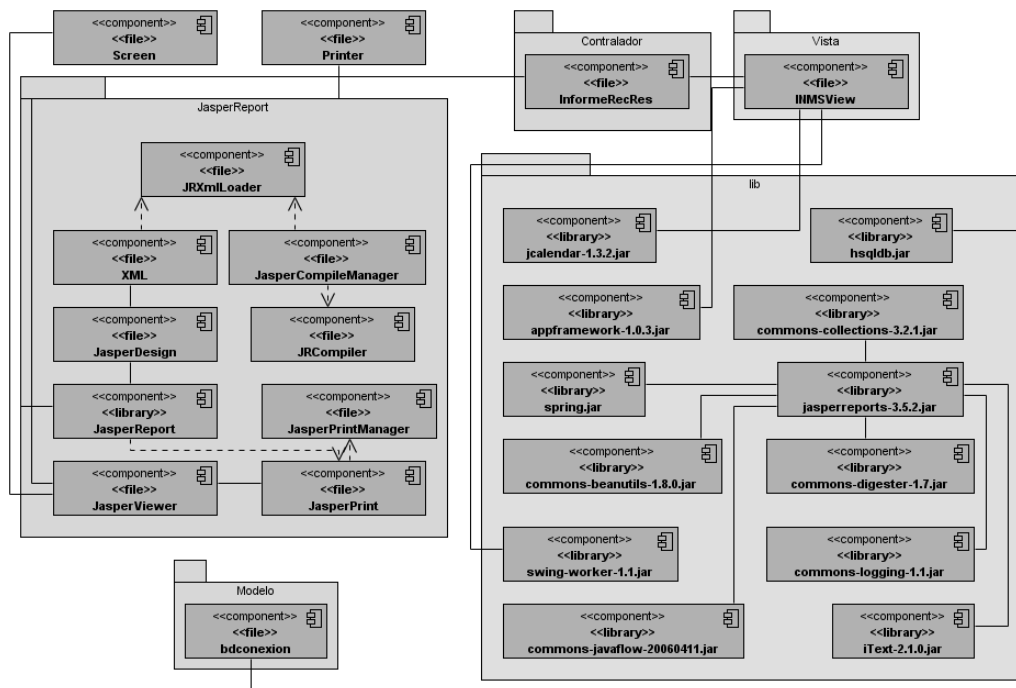


Imagen 14 Diagrama de Componentes del Caso de Uso Generar Informe Recursos y Reservas.

4.7 Pruebas del sistema propuesto.

Probar es imprescindible para verificar la calidad y el adecuado funcionamiento de un software. La prueba es un proceso de ejecución de un programa con la intención de comprobar que el producto satisface los requisitos y se comporta como se desea. Para que las pruebas tengan éxito es necesario realizar casos de pruebas que tengan probabilidad de descubrir los errores en el sistema y utilizar técnicas que nos guíen el proceso de la prueba. Para probar el sistema en cuestión se utilizaron las técnicas de caja negra.

4.7.1. Caso de Uso Generar Informe Geológico.

4.7.1.1. Descripción General.

El Caso de Uso se inicia cuando el Concesionario accede al submenú Generar Informe Geológico, del menú Generar Informes.

4.7.1.2. Condiciones de Ejecución.

El sistema debe poseer como mínimo un Informe Geológico gestionado.

4.7.1.3. Secciones para probar el Caso de Uso.

Sección	Escenarios de la Sección	Descripción de la Funcionalidad
SC 3: Generar Informe Geológico.	EC 1.1: Seleccionar un Informe para Generar el Informe Geológico con los datos de campo.	El sistema muestra un árbol con los Informes previamente registrados. El campo a seleccionar es: Informe Geológico.
	EC 1.2: Generar Informe Geológico con sus datos de campo.	El sistema verifica que exista un informe del tipo que se desea generar y que uno de ellos coincida con el Informe seleccionado.

	EC 1.3: Dejar de seleccionar un Informe o seleccionar uno incorrecto.	El sistema verifica si es seleccionado algún Informe Geológico, de no ser así muestra un mensaje de error.
	EC 1.4: Seleccionar un Informe de Recursos y Reservas y Generar un Informe Geológico.	El sistema verifica si existe un Informe Geológico con las características del Informe seleccionado, de no ser así muestra un mensaje de error.

4.7.1.4. Descripción de variable.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Informe Geológico	Selección	No	Es necesario seleccionar una opción.

4.7.1.5. Matriz de Datos.

Escenario	1	Respuesta del Sistema	Resultado	Flujo Central
Seleccionar un Informe para Generar el Informe Geológico con los datos de campo.	NA	El sistema muestra un árbol con los Informes previamente registrados. El campo a seleccionar es: Informe Geológico.		Iniciar Aplicación.
Generar Informe Geológico con sus datos de campo.	V	El sistema verifica que exista un informe del tipo que se desea generar y que uno de ellos coincida con el Informe seleccionado.		Iniciar Aplicación. Clic en "Informe Geológico". Clic en "Generar Informe Geológico".

Dejar de seleccionar un Informe o seleccionar uno incorrecto.	I	El sistema verifica si es seleccionado algún Informe Geológico, de no ser así muestra un mensaje de error.		Iniciar Aplicación. Clic en "Generar Informe Geológico".
Seleccionar un Informe de Recursos y Reservas y Generar un Informe Geológico.	I	El sistema verifica si existe un Informe Geológico con las características del Informe seleccionado, de no ser así muestra un mensaje de error.		Iniciar Aplicación. Clic en "Informe Geológico" Clic en "Generar Informe Geológico".

4.8 Conclusiones Parciales.

En este cuarto capítulo, se obtiene un sistema completamente diseñado y construido en términos de clases del diseño. De igual manera fueron sobresalientes los principios de diseño que debían adicionarse a los requisitos funcionales enumerados en el capítulo anterior. Se generaron además cada uno de los artefactos y diagramas referentes al flujo de trabajo de implementación y se realizaron pruebas al sistema desarrollado. Al concluir este capítulo se culmina la modelación completa del Inventario Nacional de Minerales Sólidos.

CONCLUSIONES GENERALES.

Concluida la investigación que trajo consigo el desarrollo del sistema informático “Inventario Nacional de Minerales Sólidos” que permitirá mantener una adecuada gestión y control administrativo de los recursos y reservas existentes en cada una de las concesiones existentes a lo largo del país y que radicará en la ONRM y Concesionarios, se pueden arribar a las siguientes conclusiones:

- ✓ Se enfatizó en la demostración de la necesidad existente en la ONRM de contar con un sistema informatizado que permitiera la gestión y control administrativo del Inventario Nacional De Minerales Sólidos. Para esto se planteó la modelación de los flujos de trabajo de negocio, requerimientos, diseño, implementación y prueba, los cuales posibilitaron un desarrollo eficiente del sistema que mejorará dicho proceso.
- ✓ Se realizó además un análisis exhaustivo de las diferentes tendencias y tecnologías actuales para desarrollar el sistema concebido de la mejor forma con las distintas herramientas existentes, enfatizando en las tecnologías de software libre. Se satisfacen también las necesidades del cliente y garantizando la construcción de una aplicación estable.
- ✓ La modelación de los distintos flujos de trabajo se realizó de forma eficiente dando la posibilidad de implementar un sistema robusto y eficiente, lo cual posibilitó la realización de una aplicación que permitirá mayor calidad cuando se gestione y administre el Inventario Nacional de Minerales Sólidos, resultado demostrado en las pruebas realizadas al sistema.

TRABAJOS REFERENCIADOS

Allen B. Tucker, Jr. 1992. *Lenguajes de Programación. 2da ed.* México : McGraw-Hill, 1992.

AmericaTI.com. 2006. The history of Java Technology. *The history of Java Technology.* [En línea] Sun Microsystem, Noviembre de 2006. <http://www.java.com/en/javahistory/>.

Asamblea, Nacional del Poder Popular. 1995. Asamblea Nacional del Poder Popular. *Oficina Nacional de Recursos Minerales.* [En línea] 1995. <http://www.onrm.minbass.cu/files/datos/Legislacion/LEGMINERA/LEYES/LEY%20MINAS/LEY-76-94%20Ley%20de%20Minas.pdf>.

2001. COPYRIGHT. *linux10.com.ar.* [En línea] 2001. <http://www.linux10.com.ar/Glosario/terminos/copyright.htm>.

Española, Real Academia. Real Academia Española / Inventario. *Real Academia Española.* [En línea] <http://buscon.rae.es/drae/>.

2009. Frameworks de aplicacion: Spring.Net. *Spring.Net.* [En línea] 17 de Diciembre de 2009. <http://www.springframework.net/>.

2001. Glosario Términos. *linux10.com.ar.* [En línea] 2001. http://www.linux10.com.ar/Glosario/terminos/software_propietario.htm.

gnu.org. 1996. La Definición de Software Libre: GNU Operating System. *GNU Operating System.* [En línea] GNU.org, 1996. <http://www.gnu.org/philosophy/free-sw.es.html>.

Jacobson, I. Booch, G. Rumbaugh, J. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison – Wesley (Edición en español por la Pearson Educación S.A, 2000.

MINBAS. 1999. *INSTRUCCION SOBRE EL BALANCE ANUAL DE LOS RECURSOS Y RESERVAS MINERALES DE LA NACION.* Ciudad de la Habana, Cuba : s.n., 1999.

—. **2009.** Oficina Nacional de Recursos Minerales / Estructura. *Oficina Nacional de Recursos Minerales*. [En línea] Noviembre de 2009. [Citado el: 13 de Diciembre de 2009.] <http://www.onrm.minbas.cu/?q=es/node/92>.

—. **2009.** Oficina Nacional de Recursos Minerales / Misión. *Oficina Nacional de Recursos Minerales*. [En línea] Noviembre de 2009. [Citado el: 13 de Diciembre de 2009.] <http://www.onrm.minbas.cu/?q=es/node/91>.

—. Resolución 215 Clasificación de los recursos y reservas de minerales sólidos. *Oficina Nacional Recursos Minerales*. [En línea] <http://www.onrm.minbas.cu>.

Minerales, Oficina Nacional de Recursos. Oficina Nacional de Recursos Minerales / Recursos y Reservas. *Oficina Nacional de Recursos Minerales*. [En línea] http://10.34.11.3:5800/portal/?q=es/recursos_y_reservas.

Stroube, Bryan. 2001. Aplicaciones de escritorio de código abierto. *Aplicaciones de escritorio de código abierto*. [En línea] 2001. <http://www.acm.org/crossroads/espanol/xrds8-4/desktopOpenSource.html>.

Stroustrup, Bjarne. 1997. "The C++ Programming Language" . s.l. : 3rd Ed. Addison-Wesley. , 1997. ISBN 0-201-88954-4 .

Sun Microsystems. 1994 - 2009. Sun Microsystems. *Sun Microsystems*. [En línea] 1994 - 2009. http://www.sun.com/emrkt/innercircle/newsletter/spain/0207spain_feature.html.

Sun Microsystems, Inc. 2008. Java: El lenguaje de Programación Java™. *Java: El lenguaje de Programación Java™*. 2008.

W3C. 2008. Guía Breve de Tecnologías XML: w3c. *w3c*. [En línea] 09 de Enero de 2008. <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.

BIBLIOGRAFÍA

Allen B. Tucker, Jr. 1992. *Lenguajes de Programación. 2da ed.* México : McGraw-Hill, 1992.

AmericaTI.com. 2006. The history of Java Technology. *The history of Java Technology.* [En línea] Sun Microsystem, Noviembre de 2006. <http://www.java.com/en/javahistory/>.

Asamblea, Nacional del Poder Popular. 1995. Asamblea Nacional del Poder Popular. *Oficina Nacional de Recursos Minerales.* [En línea] 1995. <http://www.onrm.minbass.cu/files/datos/Legislacion/LEGMINERA/LEYES/LEY%20MINAS/LEY-76-94%20Ley%20de%20Minas.pdf>.

2001. COPYRIGHT. *linux10.com.ar.* [En línea] 2001. <http://www.linux10.com.ar/Glosario/terminos/copyright.htm>.

Española, Real Academia. Real Academia Española / Inventario. *Real Academia Española.* [En línea] <http://buscon.rae.es/drae/>.

2009. Frameworks de aplicacion: Spring.Net. *Spring.Net.* [En línea] 17 de Diciembre de 2009. <http://www.springframework.net/>.

2001. Glosario Términos. *linux10.com.ar.* [En línea] 2001. http://www.linux10.com.ar/Glosario/terminos/software_propietario.htm.

gnu.org. 1996. La Definición de Software Libre: GNU Operating System. *GNU Operating System.* [En línea] GNU.org, 1996. <http://www.gnu.org/philosophy/free-sw.es.html>.

Jacobson, I. Booch, G. Rumbaugh, J. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison – Wesley (Edición en español por la Pearson Educación S.A, 2000.

MINBAS. 1999. *INSTRUCCION SOBRE EL BALANCE ANUAL DE LOS RECURSOS Y RESERVAS MINERALES DE LA NACION.* Ciudad de la Habana, Cuba : s.n., 1999.

—. **2009.** Oficina Nacional de Recursos Minerales / Estructura. *Oficina Nacional de Recursos Minerales*. [En línea] Noviembre de 2009. [Citado el: 13 de Diciembre de 2009.] <http://www.onrm.minbas.cu/?q=es/node/92>.

—. **2009.** Oficina Nacional de Recursos Minerales / Misión. *Oficina Nacional de Recursos Minerales*. [En línea] Noviembre de 2009. [Citado el: 13 de Diciembre de 2009.] <http://www.onrm.minbas.cu/?q=es/node/91>.

—. Resolución 215 Clasificación de los recursos y reservas de minerales sólidos. *Oficina Nacional Recursos Minerales*. [En línea] <http://www.onrm.minbas.cu>.

Minerales, Oficina Nacional de Recursos. Oficina Nacional de Recursos Minerales / Recursos y Reservas. *Oficina Nacional de Recursos Minerales*. [En línea] http://10.34.11.3:5800/portal/?q=es/recursos_y_reservas.

Stroube, Bryan. 2001. Aplicaciones de escritorio de código abierto. *Aplicaciones de escritorio de código abierto*. [En línea] 2001. <http://www.acm.org/crossroads/espanol/xrds8-4/desktopOpenSource.html>.

Stroustrup, Bjarne. 1997. "The C++ Programming Language" . s.l. : 3rd Ed. Addison-Wesley. , 1997. ISBN 0-201-88954-4 .

Sun Microsystems. 1994 - 2009. Sun Microsystems. *Sun Microsystems*. [En línea] 1994 - 2009. http://www.sun.com/emrkt/innercircle/newsletter/spain/0207spain_feature.html.

Sun Microsystems, Inc. 2008. Java: El lenguaje de Programación Java™. *Java: El lenguaje de Programación Java™*. 2008.

W3C. 2008. Guía Breve de Tecnologías XML: w3c. *w3c*. [En línea] 09 de Enero de 2008. <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.

GLOSARIO DE TÉRMINOS

-A-

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

AWT: Es el API estándar para proporcionar interfaces gráficas de usuario (GUIs) para programas Java.

-B-

Bytecodes: Código intermedio más abstracto que el código máquina.

-C-

C: Es un lenguaje de programación de nivel medio ya que combina los elementos del lenguaje de alto nivel con la funcionalidad del ensamblador.

C++: Es un lenguaje imperativo orientado a objetos derivado del C.

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

Concesionario: Son las personas naturales o jurídicas, debidamente autorizadas para el ejercicio de una o varias fases de la actividad minera por el correspondiente título.

-G-

Geofísica: Es la ciencia que se encarga del estudio de la tierra desde el punto de vista de la física.

Geología: La geología es la ciencia que estudia la forma interior del globo terrestre, la materia que lo compone, su mecanismo de formación, los cambios o alteraciones que éstas han experimentado desde su origen y la colocación que tienen en su actual estado.

-I-

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 GUI.

-J-

JDK: Juego de Herramientas de Desarrollo de Software.

JRE: Entorno de ejecución de Java.

-K-

Kernel: El kernel ó núcleo de linux se puede definir como el corazón de este sistema operativo.

-S-

Sun Microsystems: Empresa líder en soluciones globales con productos y servicios de alta Tecnología Informática para sistemas abiertos.

Swing: Es una biblioteca gráfica para Java que forma parte de las Java Foundation Classes (JFC). Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, despletables y tablas.

-W-

Widget: En informática, un *widget* es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

Anexo 1 Diagramas de Actividades del Negocio.

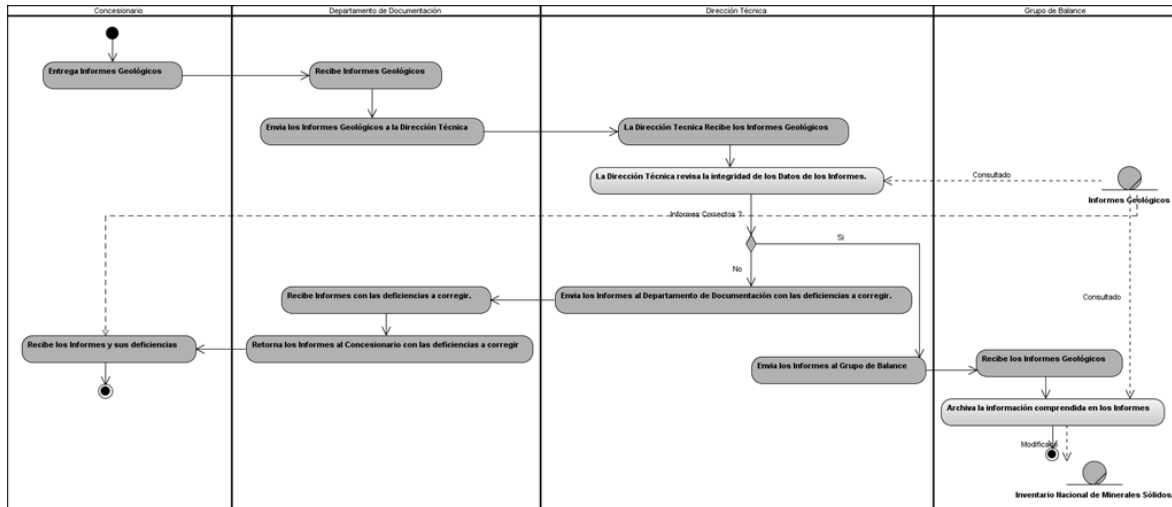


Imagen 15 de Actividades Caso de Uso Entregar Informes Geológicos.