

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 6



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

Título:

“Herramienta para la gestión de modelos de bloques geológicos”

Autora: Sucler María Mojena Sánchez

Tutor: Ing. Eddy Dangel Quesada Rodríguez

La Habana, junio de 2013

“Año 55 de la Revolución”

DECLARACIÓN DE AUTORÍA:

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año 2013.

Sucel María Mojena Sánchez

Ing. Eddy Dangel Quesada Rodríguez



Un hombre como yo no es descriptible por las cosas que dice o los sentimientos que tiene, sino por lo que piensa y cómo lo piensa.

Albert Einstein

DATOS DE CONTACTO:

Tutor: Ing. Eddy Dangel Quesada Rodríguez.

Breve Currículum:

Ing. Eddy Dangel Quesada Rodríguez graduado de Ingeniero en Ciencias Informáticas en el 2008. El mismo tiene la categoría docente de Instructor y cuenta con 6 años de experiencia en proyectos productivos. Actualmente es líder del proyecto de Sistema Minero Cubano.

Correo Electrónico: edquezada@uci.cu

AGRADECIMIENTOS

A la persona que me trajo al mundo, ha cultivado mi ser y ha sido y es mi mamá y papá a la vez Clara Sánchez Torres.

A mi hermano Alexis Medina por existir y estar siempre a mi lado en todas las etapas de mi vida y brindarme unos sobrinos maravillosos (Neli, Jeisser y Jeisse).

A todos mis seres queridos que de una forma u otra estuvieron implicados y marcaron mi futuro.

A mi novio Antonio por brindarme su cariño, amor, comprensión, dedicación, confianza y por estar siempre que lo necesito.

A mi tutor por todo su apoyo, exigencia y críticas constructivas con el objetivo de lograr exitosos resultados en la investigación.

A mi tribunal de tesis por sus precisas acotaciones que le dieron a este trabajo mayor calidad y rigor científico.

A la vida por darme la oportunidad de conocer personas maravillosas que me han ayudado a crecer, a ser una mejor persona cada día y a entender que lo importante es realmente invisible a los ojos.

A ti que estás leyendo estas líneas.

DEDICATORIA

A la persona más especial del mundo, Clara Sánchez.

A mi hermano incondicional Alexis Medina.

*A mis sobrinos del corazón presentes y no presentes en la vida, pero si en mi ser por el resto de
mis días.*

A mi novio Antonio por tanto amor y dedicación.

A mi hermosa familia comprendidas por mis tías Enriqueta y Justina.

RESUMEN

Durante largos años de estudios e investigaciones, las distintas entidades geológicas del país han desarrollado la explotación de recursos minerales existentes en la corteza terrestre. Para lograr esto se realiza un estudio de los distintos procesos que ocurren hoy en día en la gestión de los datos geológicos contenidos en un yacimiento mineral, utilizando una de las técnicas de modelado más empleada por geólogos y mineros, la modelación geológica regional. El trabajo que a continuación se expone, versa sobre el completo desarrollo de un sistema informático como herramienta para la gestión de datos asociados a un yacimiento comprendido en modelos de bloques. Para comprender la necesidad existente en la actualidad, se realizó un exhaustivo estudio de los distintos procesos que rodean a las entidades cubanas que se encargan de gestionar modelos de bloques geológicos. Luego de analizada la situación actual se adentra en la construcción de la aplicación propuesta, para lo cual se escogieron un conjunto de herramientas y tecnologías orientadas al software libre. La metodología de desarrollo de software utilizada guió la elaboración del producto, el cual recorrió todos los flujos de trabajo propuestos por la misma. Al culminar la fase de construcción se obtuvo un software calificado para dar cumplimiento a los procesos donde existían deficiencias. Dicho software fue sometido a distintos tipos de pruebas, obteniendo en todas los resultados esperados.

Palabras claves: geología, minería, modelación, modelo de bloques, recursos minerales.

ABSTRACT

During long years of study and research, the different entities of the country have developed geological exploitation of mineral resources existing in the earth's crust. To achieve this is a study of the various processes occurring today in the management of geological mineral in a deposit using one of the most widely used modeling techniques for geologists and miners, the regional geological modeling. The work discussed below, concerns the development of a complete computer system as a tool for managing data associated with a field covered by block models. To understand the need that exists today, was made an exhaustive study of the various processes surrounding Cuban entities are responsible for managing geological block models. After analyzing the current situation goes into the construction of the proposed application, for which we chose a set of tools and technologies for the free software. The software development methodology used led product development, which flows through all the work proposed for the same. Upon completion of the construction phase was obtained qualified software to comply with the processes where deficiencies exist. This software was subjected to various types of evidence, obtaining in all the expected results.

Keywords: block model, geology, mineral resources, mining, modeling.

ÍNDICE

INTRODUCCIÓN:.....1

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN.6

 1.1 Conceptos asociados al dominio del problema.....6

 1.1.1 Yacimiento Mineral.....6

 1.1.2 Proceso Minero.....8

 1.2 Modelación geológica tridimensional9

 1.3 Modelo de bloques11

 1.4 Pasos a tener en cuenta para realizar la herramienta informática.13

 1.5 Análisis de los principales software existentes.14

 1.6 Conclusiones Parciales15

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS ACTUALES......16

 2.1 Lenguajes de Programación.....16

 2.1.2 C#16

 2.1.3 C/C++17

 2.2 Frameworks.....18

 2.2.2 Framework Qt18

 2.2.3 Visualization Toolkit (VTK).19

 2.3 Integrated development environment (IDE).19

 2.3.1 Visual Studio 201020

2.3.2 QtCreator	20
2.4 Sistema Gestor de Base de Datos (SGBD).	21
2.4.1 PostgreSQL.	22
2.4.2 SQL Server	22
2.5 Metodologías de Desarrollo.	23
2.5.1 Rational Unified Process (RUP).	24
2.5.2 Agile Unified Process (AUP).	24
2.6 Herramientas CASE.	25
2.6.1 Visual Paradigm.	26
2.6.2 Rational Rose.	26
2.7 Lenguaje Unificado de Modelado (UML).....	27
2.8 Herramientas y tecnologías seleccionadas para la investigación.....	28
2.9 Conclusiones Parciales.	29
CAPÍTULO 3: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	30
3.1 Modelo de Dominio.....	30
3.1.2 Diagramas de clases del dominio.....	30
3.1.3 Descripción del modelo de dominio.....	31
3.2 Requisitos funcionales y no funcionales del sistema.	31
3.3 Descripción del sistema propuesto.	34
3.3.1 Descripción de los actores.	35

3.3.2 Casos de Uso del Sistema.....	35
3.3.3 Diagrama de Caso de Uso del Sistema.....	35
3.3.4 Descripción textual de los Casos de Uso del Sistema.....	36
3.4 Modelo de Diseño.....	42
3.4.1 Diagrama de clases del diseño.....	42
3.5 Diseño de la Base de Datos.	43
3.5.1 Diagrama Entidad Relación.	43
3.6 Arquitectura del Sistema.....	44
3.7 Patrones de Diseño.....	45
3.8 Conclusiones Parciales.	46
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	47
4.1 Implementación.....	47
4.1.1 Estándares de codificación.	47
4.1.2 Tratamiento de errores.....	48
4.1.3 Modelo de implementación.	49
4.1.4 Modelo de Despliegue.	50
4.2 Prueba del sistema propuesto.	50
4.2.1 Prueba del sistema.	50
4.3 Caso de estudio.	52
4.4 Beneficios del sistema.....	57

4.5 Conclusiones Parciales.....	57
CONCLUSIONES GENERALES.....	58
RECOMENDACIONES.....	59
REFERENCIAS BIBLIOGRÁFICAS.....	60
BIBLIOGRAFÍA.....	62
GLOSARIO.....	65

ÍNDICE DE FIGURAS

Figura 1: Etapas del proceso minero.....8

Figura 2: Fases del proceso minero cubano.9

Figura 3: Modelación geológica estructural.....10

Figura 4: Modelación geológica regional.11

Figura 5: Modelo de bloques geológico.11

Figura 6: Modelo de bloques con leyenda de colores.12

Figura 7: Diagrama de dominio.30

Figura 8: Diagrama de Caso de Uso del Sistema.36

Figura 9: Diagrama de Diseño del Caso de Uso Gestionar Proyecto.....42

Figura 10: Modelo Entidad Relación.....43

Figura 11: Ejemplo de tratamiento de errores a nivel de interfaz.48

Figura 12: Diagrama de despliegue.....50

Figura 13: Diseño del caso de prueba Nuevo Modelo de Bloques.....51

Figura 14: Crea un nuevo proyecto.53

Figura 15: Crear un nuevo modelo de bloques.....53

Figura 16: Atributos del modelo.....54

Figura 17: Importar datos del modelo.....55

Figura 18: Visualización 3D.56

ÍNDICE DE TABLAS

Tabla 1: Herramientas y Tecnologías escogidas.	29
Tabla 2: Actores del sistema.....	35
Tabla 3: Descripción CUS Gestionar Proyecto.....	36
Tabla 4: Diagrama de Arquitectura del Sistema.	44
Tabla 5: Iteraciones de prueba de caja negra.....	52
Tabla 6: Datos básicos del modelo de bloques.	52
Tabla 7: Atributos añadidos al modelo	55

INTRODUCCIÓN:

La explotación de los recursos minerales existentes en la corteza terrestre ha sido una de las actividades más antiguas realizadas por el hombre, la cual influyó positivamente en su desarrollo económico y social. El hombre a lo largo de los años, ha mejorado las técnicas utilizadas para la explotación de minerales, porque en los primeros tiempos la labor geológica era llevada a cabo por personas, con mayores o menores conocimientos acerca del tema. El desarrollo industrial alcanzado por el hombre a partir de 1830, aceleró los procesos de producción y explotación de recursos minerales, algo fundamental en la industria moderna. Tradicionalmente la comprensión de las estructuras geológicas se ha realizado mediante técnicas bidimensionales, pero los resultados no son óptimos en todas sus dimensiones y esta es una de las limitaciones existentes. Debido a la realidad tridimensional de las estructuras geológicas, su solución y comprensión pasa, inevitablemente, por una aproximación tridimensional que minimice los posibles errores en la proyección.

La modelación geológica tridimensional (HOULDING, 1994), ha permitido a los científicos de las ramas asociadas a las Geociencias comprender los principales fenómenos que ocurren en el subsuelo de la corteza terrestre y que no son apreciables a simple vista. Las diferentes técnicas de visualización tridimensional permiten observar las zonas mineralizadas, superficie topográfica y otras estructuras geológicas contenidas en un yacimiento mineral. Este enfoque crea una única relación entre la interpretación de los datos geológicos y la construcción de modelos. Además este tipo de modelación es una de las actividades más realizadas por geólogos y mineros. La modelación de un yacimiento mineral permite conocer sus principales parámetros geotécnicos y geofísicos, que a la vez interpreta la geometría de las estructuras y profundidades de la misma.

En la década del ochenta del siglo XX en Cuba, se utilizaban los métodos tradicionales de forma manual para la estimación de recursos minerales, por lo que proporcionaba un alto grado de incertidumbre y todo se iba deteriorando con el tiempo, por lo que debía perfeccionarse, migrando toda esta información que se encontraba en formato duro al digital. Empresas cubanas a partir de 1995 realizaron análisis del flujo de datos en todas las minas de níquel. Luego se proporcionó la extensión de la tecnología con el sistema

Microniq, que sigue en vigencia actualmente y de esta forma se automatiza la información relacionado a las minas en los yacimientos minerales. Los sistemas informáticos cubanos desarrollados hasta ese momento, contaban con una débil representación tridimensional de la información asociada a los objetos geológicos, lo que impedía a los especialistas mineros realizar un análisis profundo y real de esta información. El auge que cobraba la industria minera cubana y la necesidad de contar con sistemas para el control de los recursos minerales de los yacimientos, fue el factor decisivo que posibilitó la adquisición de los productos informáticos privativos imperantes en el mundo minero internacional.

A partir del año 2006 el Ministerio de la Industria Básica (MINBAS) comenzó las negociaciones con compañías extranjeras para adquirir los sistemas informáticos Gems, Surpac y Datamine con el objetivo de garantizar la fiabilidad de la estimación de recursos minerales en algunas de las empresas mineras cubanas. Actualmente existen empresas mineras cubanas que no cuentan con el presupuesto necesario para comprar sistemas informáticos mineros extranjeros, que realicen el cálculo de los recursos minerales, el control y planificación de la actividad minera. Algunas de las aplicaciones informáticas compradas por empresas cubanas no permiten el acceso concurrente a las datos mineros lo cual hace engorroso el proceso de gestión de los mismos.

El desarrollo de los sistemas automatizados en este campo ha permitido un incremento en la calidad de las estimaciones en diferentes tipos de yacimientos, pero aún existen deficiencias en cuanto a la precisión y exactitud de los cálculos, uno de ellos es conocer la ubicación del bloque de estimación. En cuanto a la seguridad de la información y su acceso, se puede decir que la misma no perdura en el transcurso del tiempo. Por tal motivo es necesaria la persistencia de los datos de un modelo de bloques, este es un factor irreversible en la decisión de llevar a cabo una inversión minera. Las condiciones de trabajo que existen no son las más idóneas, ya que cuentan con una tecnología informática que no se adapta a las condiciones actuales y conduce a obtener resultados no deseados.

Encontrar un método factible que permita lograr un mejor aprovechamiento de los recursos minerales es una necesidad. De esta forma es preciso tener un modelo de bloques donde sobre él se puedan realizar actividades mineras como: la planificación minera, explotación de minerales y estimación de recursos.

Para la selección de éste se han utilizado modelos y métodos que poseen criterios de confiabilidad y seguridad.

Debido a las deficiencias mencionadas anteriormente surge el siguiente **problema a resolver** que da inicio a esta investigación: ¿Cómo garantizar la persistencia de los datos asociados a un yacimiento mineral utilizando un modelo de bloques geológicos?

El problema planteado se enmarca en el **objeto de estudio**: Los procesos de gestión de datos asociados a un yacimiento mineral.

Delimitado por el **campo de acción**: Los procesos de gestión de datos asociados a un yacimiento mineral utilizando modelos de bloques geológicos.

Para dar solución al problema planteado se define como **objetivo general**: Implementar una herramienta informática que gestione los datos asociados a un yacimiento mineral contenidos en modelos de bloques geológicos utilizando un sistema gestor de base de datos.

La idea a defender es: Si se implementa una herramienta informática que gestione los datos asociados a un yacimiento mineral contenidos en modelos de bloques geológicos utilizando un sistema gestor de base de datos, entonces se logrará la persistencia de los mismos.

Para alcanzar el objetivo propuesto, se realizarán las siguientes **tareas de la investigación**:

- Documentar la estructura y características de un modelo de bloques geológicos.
- Caracterizar las principales herramientas informáticas y otras soluciones existentes que realizan la gestión de modelos de bloques geológicos.
- Definir las herramientas y tecnologías que se utilizarán para implementar la herramienta informática.
- Elaborar la documentación técnica necesaria asociada a la herramienta que se implementará.
- Implementar la herramienta para la gestión y manipulación de modelos de bloques geológicos.
- Realizar las pruebas a la herramienta informática.

Para lograr un mejor entendimiento acerca de los problemas planteados anteriormente, se utilizan los siguientes métodos científicos.

Teóricos:

- **Análisis Histórico Lógico:** Este método posibilitó que se realizara un estudio acerca de la evolución del proceso de gestión de datos geológicos de los yacimientos minerales cubanos. También posibilitó un análisis histórico sobre los sistemas informáticos que se han utilizado para la gestión de datos geológicos empleando modelo de bloques, permitiendo conocer todos sus parámetros geotécnicos.
- **Analítico-Sintético:** Este método permitió realizar un análisis de las bibliografías disponibles para sintetizar el conocimiento necesario para desarrollar la presente investigación. Además posibilitó definir los principales conceptos y analizar soluciones existentes tanto nacionales como internacionales.

Empíricos:

- **Observación:** Este método permitió analizar diferentes software mineros, con el objetivo de conocer como realizan el modelo de bloques geológicos. Definiendo de este modo la necesidad de la investigación, creando una aplicación para la gestión de modelos de bloques geológicos y obtener resultados de la misma.

Resultados esperados:

1. Una herramienta informática para la gestión de modelos de bloques geológicos.
2. La documentación teórica asociada al desarrollo de la herramienta anterior.

El trabajo de diploma consta de los siguientes **capítulos**:

Capítulo 1. Fundamentos Teóricos de la Investigación: Incluye conceptos asociados al dominio del problema, necesarios para el entendimiento correcto de la solución que se propone. Hace referencia al estado del arte de soluciones existentes desde el ámbito nacional e internacional.

Capítulo 2. Herramientas y tecnologías: Se realiza un análisis de las principales herramientas, tecnologías y metodologías que se utilizan para el desarrollo de sistemas informáticos, así como la selección de algunas de estas para el desarrollo de la solución propuesta.

Capítulo 3. Descripción de la propuesta de solución: Se realiza el análisis del sistema, se define la arquitectura del sistema y los patrones de diseño para asegurar de esta forma las buenas prácticas en la programación de la solución. Además se logra encapsular los requisitos funcionales del sistema para facilitar de esta forma la implementación del sistema.

Capítulo 4. Implementación y prueba: Constituye la última fase de la investigación, esta presenta los distintos componentes que conforman el sistema y se realiza las pruebas del sistema a través del método de pruebas de caja negra.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN.

La modelación geológica tridimensional es una de las formas de representación espacial de un yacimiento mineral, pero también se puede apreciar sobre el mismo sus límites y estructuras internas, lo cual genera un gran volumen de datos. Aspectos como estos se explicarán en el desarrollo de este capítulo. Además se necesita tener conocimiento acerca de las principales técnicas de modelado, debido a que la explotación minera es una de las actividades fundamentales realizadas en el país y es de vital importancia económica para el mismo.

1.1 Conceptos asociados al dominio del problema

Existen un conjunto de conceptos que están muy relacionados con la investigación, razón por la cual a continuación se explica de forma concisa en qué consisten.

1.1.1 Yacimiento Mineral

Los yacimientos minerales son acumulaciones naturales de un mineral que permiten su explotación con rendimiento económico. Se pueden distinguir tres grandes grupos de yacimientos minerales, los magmáticos¹, sedimentarios² y metamórficos³. La ley de minas plantea, que un yacimiento mineral: *“Es cualquier acumulación natural de sustancias minerales en el suelo o en el subsuelo, que pueda ser utilizado y explotado como fuente de materia prima y como fuente de energía, y las concentraciones de piedras preciosas y semipreciosas y de cualquier otra sustancia mineral cuya explotación tenga importancia económica”* (ALARCÓN, 1994).

Durante numerosas ocasiones se encuentra en el tratamiento de los recursos y reservas minerales la terminología exploración, lo cual resulta de vital importancia conocer su significado, a continuación se hace referencia al artículo 3 del capítulo 1 de la Ley de Minas que lo conceptualiza: *“La exploración es un*

¹ Proceso de formación de los magmas hasta su solidificación y formación de rocas.

² Parte de la superficie terrestre donde se acumulan minerales.

³ Depósitos minerales sometidos a presiones y temperaturas elevadas.

conjunto de operaciones, trabajos y labores mineras que tienen como objetivo la determinación de la estructura del yacimiento, la morfología, dimensiones y condiciones de yacencia del cuerpo mineral, la tectónica de la zona que lo contiene, el contenido y calidad del o de los minerales existentes en el mismo, así como el cálculo de las reservas, incluyendo la evaluación económica del yacimiento y otros estudios que ayuden a su mejor explotación” (ALARCÓN, 1994) .

La explotación de minerales ocurre de carácter natural asociados a procesos y ambientes geológicos específicos según su tipo. La Ley de Minas plantea: *“La explotación es conjunto de operaciones, obras, trabajos y labores mineras destinadas a la preparación y desarrollo del yacimiento y a la extracción y transportación de los minerales” (ALARCÓN, 1994)*. Resulta también de un gran significativo conocer el significado de ley de mineral en numerosas ocasiones, a continuación se hace referencia al artículo de la Ley de Minas que lo conceptualiza: *“Ley del Mineral es una concentración de metal contenido en una mena” (ALARCÓN, 1994)*. Una mina es un yacimiento en el cual es posible realizar la extracción de un mineral con beneficios económicos. En la Gaceta Oficial de la República de Cuba se traza el siguiente concepto: *“Mina es una obra resultante del conjunto de excavaciones e instalaciones superficiales y subterráneas que se realizan para la investigación y la explotación de un yacimiento mineral” (ALARCÓN, 1994)*.

Los minerales son una sustancia de origen natural y pueden ser de uso en la actividad humana de manera directa o indirectamente. En la Ley de Minas se plasma que: *“Un mineral es sustancia inorgánica que se halla en el suelo o en el subsuelo, y principalmente aquella cuya explotación ofrece interés económico” (ALARCÓN, 1994)*. Este recurso es de gran importancia para la economía de un país y resultan esenciales para la vida. Se dividen en tres grandes grupos, los metálicos, los no-metálicos y los energéticos. Los metálicos son minerales de mena de los cuales se obtienen metales de interés. Los no-metálicos son minerales empleados directamente en procesos industriales, tecnológicos y de construcción. Por último los recursos energéticos, por mencionar algunos de ellos: petróleo, gas natural, carbón y uranio.

Conocer los minerales que contiene una mina es sumamente significativo y se denominan mena. La Ley de Minas lo conceptualiza como: *“Mena es porción útil de un mineral metalífero” (ALARCÓN, 1994)*.

Además la exploración de yacimientos es denominada prospección, y la Ley de Minas la define como: *“Conjunto de trabajos con empleo de técnicas y métodos que tienen como objetivo la búsqueda de indicios de concentraciones minerales que pudieran constituir yacimiento”* (ALARCÓN, 1994). Toda esta búsqueda es en una zona la cual es mineralizada, esto constituye algo que no se debe dejar de mencionar, la Gaceta lo conceptualiza como: *“Zona mineralizada es aquella extensión del suelo o subsuelo en la que se encuentran concentraciones de mineral de aprovechamiento económico”*(ALARCÓN, 1994).

1.1.2 Proceso Minero

El proceso minero está formado por las etapas mostradas en la **Figura 1**. La prospección que comienza con la definición de los blancos a buscar. Le sigue la exploración, esta se encarga de afirmar las hipótesis planteadas en la etapa de prospección. Luego está el desarrollo técnico en la que se desarrolla la ingeniería del proyecto. Posteriormente se efectúa la operación de la mina que consiste en la extracción y clasificación de los minerales. Otras de las etapas cruciales en el proceso es el transporte de minerales, para que el mineral pueda ser comercializable se recurren a métodos de beneficio de minerales, siendo esta otra etapa del proceso. La última etapa es el cierre de mina, donde se ejecutan acciones que garantizan el cierre de la mina, asegurando la sostenibilidad de las comunidades y el medio ambiente (CASTELLS, 2012).

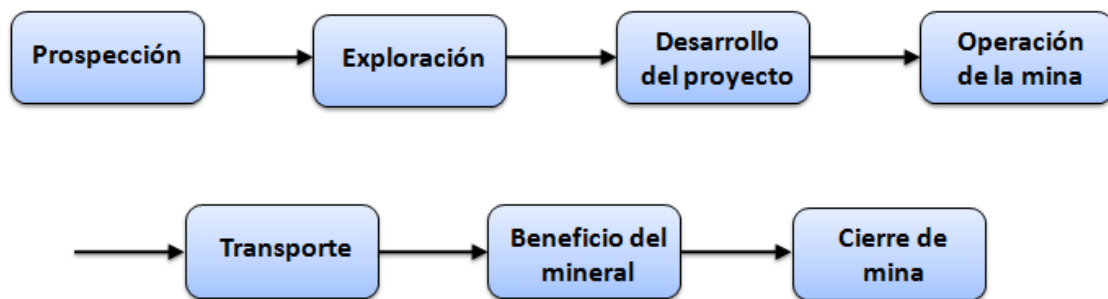


Figura 1: Etapas del proceso minero.

En Cuba las fases de la minería son la prospección, la exploración y la explotación, ver **Figura 2** . La prospección se basa en estudios que se le hacen a la zona, tomando de estas muestras. Luego se determina la presencia de un yacimiento a través de un método muy importante que son las perforaciones, incidiendo está en la etapa de exploración. De su estudio y valoración depende la viabilidad económica y proponerse la tercera fase de las actividades mineras, la explotación, implica actividades que van desde la extracción hasta la comercialización del producto. En un artículo publicado en la Revista de Minería y Geología, (IZNAGA, 2001) plantea que: *“La exploración de minerales es una actividad altamente costosa y creativa; además de los modelos tecnológicos que requiere, ella descansa en gran medida en la adquisición y uso de la información geológica y hasta de factores económicos que influyen en el proceso de la exploración”*.

Todas estas actividades realizadas sobre una mina, forman un producto final, asegurando la calidad y estabilidad de la zona minera.



Figura 2: Fases del proceso minero cubano.

Un modelo fundamental para la minería es el modelo de bloques, el cual está presente en las siguientes etapas del proceso minero como: prospección, exploración y explotación.

1.2 Modelación geológica tridimensional

La modelación geológica tridimensional ha alcanzado un papel fundamental dentro de las diferentes ramas de las Geociencias. Para geólogos y mineros es imprescindible modelar tridimensionalmente las diferentes estructuras geológicas de un yacimiento mineral. La modelación geológica de un yacimiento mineral se realiza con el fin de representar geoméricamente las estructuras geológicas que componen un

yacimiento mineral. Existen diversas técnicas de modelado de yacimiento minerales, es decir un modelo geológico consta de un modelo estructural, un modelo sedimentario-estratigráfico, un modelo litológico y un modelo regional. El modelo estructural es una de las diferentes técnicas de visualización tridimensional, mostrada en la **Figura 3**. Mediante la cual se pueden observar las zonas mineralizadas, superficie topográfica y otras estructuras geológicas contenidas en un yacimiento mineral. En (NING, 2011) definen la modelación 3D como: "... *sección estratigráfica en las conexiones de los bloques adyacentes...*".

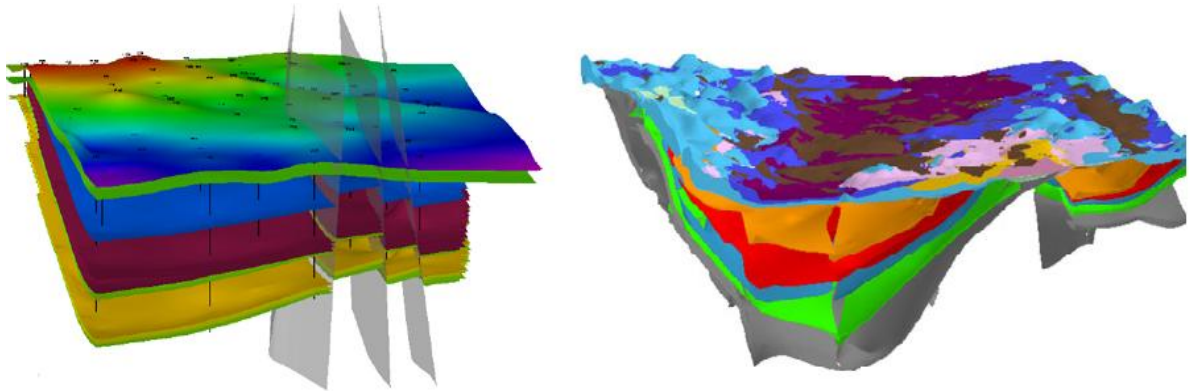


Figura 3: Modelación geológica estructural.

El modelo regional, expuesto en la **Figura 4**, consiste en dividir un yacimiento mineral en celdas tridimensionales, llamados bloques. En cada bloque del modelo se registran los principales atributos geotécnicos del yacimiento mineral, por ejemplo: las concentraciones de minerales, el tipo de roca y el valor económico del bloque. La idea que existe detrás de este modelo, es que el volumen geológico puede ser cortado en uno o más volúmenes de elementos base (CHEN, 1996).

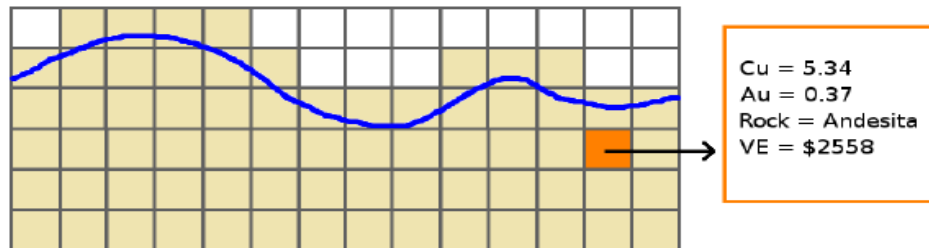


Figura 4: Modelación geológica regional.

1.3 Modelo de bloques

El modelo de bloque es una de las técnicas de representación tridimensional más utilizadas para la modelación de un depósito mineralizado, ver **Figura 5**. Esta técnica de modelado se basa en dividir el yacimiento mineral en celdas tridimensionales y registrar en cada una de ellas los principales atributos geotécnicos del yacimiento. En la mayoría de los casos las celdas tridimensionales o bloques tienen forma de paralelepípedos. A la hora de crear un modelo de bloque, se debe tener en cuenta que este se debe adaptar a la morfología del yacimiento mineral. La mayoría de las estimaciones de los recursos se llevan a cabo usando este método de modelado de bloques, ya sea como modelos de malla ([DUKE, 2000](#)).

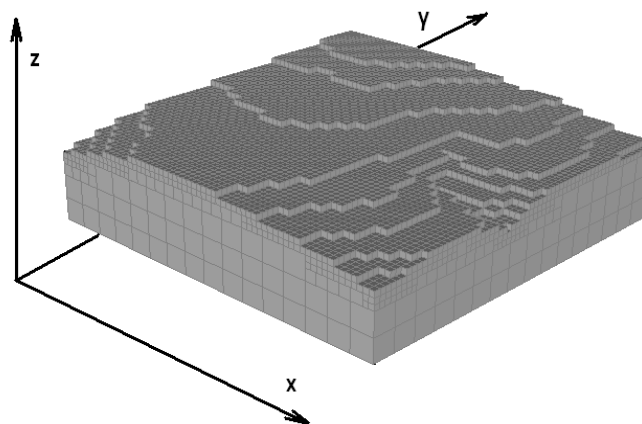


Figura 5: Modelo de bloques geológico.

Este modelo proporciona las siguientes actividades por mencionar algunas de ellas como: la estimación de recursos, la planificación minera, el control de leyes, el reporte de tonelaje, el diseño de cantera, entre otros. Desde una perspectiva computacional (RUBIO, 2008) plantea que: “Un modelo de bloques es un arreglo de n volúmenes (cubos) distribuidos en el espacio”. El conjunto completo de volúmenes representa la geometría global de un depósito o región determinada. Este modelo es una forma de base de datos espacial de referencia que proporciona un medio para modelar tridimensionalmente un depósito, tales como pozos de perforación. Este es un método que permite aplicarles leyendas de colores a la visualización a partir de los valores de un atributo del bloque, ver **Figura 6**. Además admite la visualización tridimensional y bidimensional a través de planos transversales.

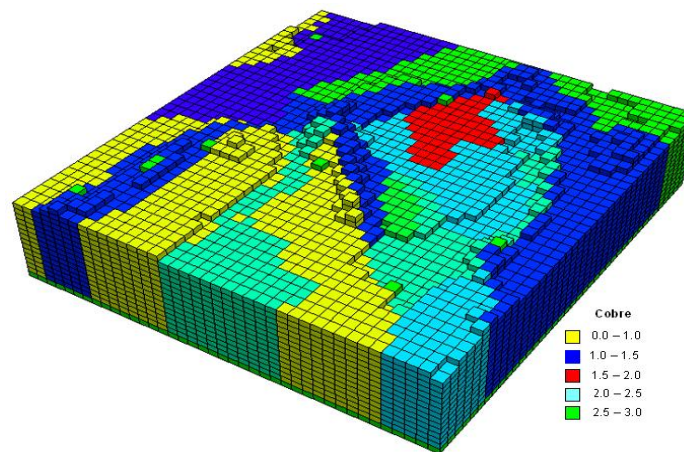


Figura 6: Modelo de bloques con leyenda de colores.

En la ayuda del software Surpac se propone el siguiente flujo de trabajo para el modelo de bloques.

1. Determinar la geometría del modelo, sus dimensiones, su coordenada de origen y crear el modelo vacío.
2. Determinar los atributos iniciales del modelo, especificar el tipo de dato de cada atributo y si el mismo es de carácter obligatorio o no. (Por ejemplo oro, plata, tipo de roca, entre otros.)

3. Crear las restricciones estructurales del modelo a partir de objetos geométricos. (Por ejemplo superficie de terreno, superficies de las capas litológicas, líneas de los contornos de zonas mineralizadas.)
4. Aplicar las restricciones al modelo y asignar códigos que denoten los diferentes tipos de rocas, menas y otros materiales a los atributos del modelo.
5. Verificar visualmente si las restricciones aplicadas han producido un resultado deseado.
6. Llenar los atributos de los bloques del modelo con valores numéricos utilizando varios métodos de llenado y aplicando las restricciones estructurales para estimar los valores de los bloques que están contenidos dentro de estas zonas.
7. Verificar los resultados de proceso de llenado del modelo utilizando métodos disponibles como gráficos tridimensionales, secciones transversales y reportes.

1.4 Pasos a tener en cuenta para realizar la herramienta informática.

En la presente investigación se analizaron diferentes aplicaciones informáticas que realizan la modelación geológica utilizando la técnica de modelo de bloques y cada cual define sus propios pasos para la creación del modelo. Los pasos que se emplearán en la solución propuesta de la presente investigación son los siguientes:

1. Crear el modelo vacío, su coordenada de origen y sus dimensiones.
2. Determinar los atributos iniciales del modelo y especificar el tipo de dato de los mismos.
3. Verificar visualmente si las restricciones aplicadas han producido un resultado deseado.
4. Aplicar las restricciones estructurales para estimar los valores de los bloques que están contenidos dentro de estas zonas.
5. Verificar los resultados de las restricciones aplicadas métodos disponibles como gráficos tridimensionales.

1.5 Análisis de los principales software existentes.

En Cuba los sistemas informáticos desarrollados para la minería, no crean, ni manipulan modelo de bloques, por tanto es necesario el uso de sistemas extranjeros. Actualmente existen varios software a nivel mundial dedicados a la gestión de los procesos geológicos mineros, dentro de ellos se destacan Gems, Surpac, Datamine y Vulcan. Muchas de las empresas cubanas han adquirido licencias de estos sistemas informáticos para certificar la calidad de la estimación de los recursos y reservas de yacimientos minerales cubanos, y poder utilizar capital extranjeros para financiar sus proyectos mineros. Aunque estas aplicaciones gocen de gran prestigio internacional, el alto precio de costo de sus licencias provoca que la mayoría de las empresas cubanas que realizan la pequeña minería puedan adquirirlos.

Todas las aplicaciones informáticas mencionados anteriormente, tienen módulos que se dedican a la creación y manipulación de modelos de bloques geológicos. Estas características les permiten a los usuarios:

- La estimación de los recursos minerales de un yacimiento mineral utilizando diversos algoritmos de interpolación espacial.
- La planificación de la producción minera a corto, mediano y largo plazo.
- Reportar el tonelaje de los recursos minerales del yacimiento y su clasificación en medidos, indicados e inferidos.
- La optimización del diseño de la cantera.
- La visualización tridimensional de las variaciones de las diferentes propiedades del modelo a través de la aplicación de leyendas de colores.
- La visualización bidimensional de diversas zonas del modelo a partir de secciones transversales del mismo.

Aunque todos estos sistemas realizan la gestión de modelos de bloques geológicos, los métodos utilizados son diferentes. Todos ellos emplean diversas técnicas de subdivisión espacial para la creación de los sub bloques del modelo, por citar un ejemplo el Surpac divide los bloques en octantes y el Datamine lo divide en dos. En el Datamine, Surpac y Vulcan el origen de coordenadas del modelo se encuentra en el vértice inferior izquierda de la caja envolvente del modelo mientras que en el Gems se encuentran en el vértice superior izquierdo de la caja envolvente del modelo. Todos estos sistemas permiten la importación y exportación de modelos de bloques en diferentes formatos pero no existe un estándar único para el formato de estos ficheros, cada software define su propio estándar. La persistencia de los datos de los modelos de bloques generados por estos sistemas se logra mediante ficheros, binarios y de texto aunque sistemas como el Gems, Surpac y Vulcan utilizan además de los ficheros sistemas gestores de bases de datos, específicamente el MS Access. El tipo de SGDB utilizado por estos sistemas no tiene una arquitectura cliente-servidor lo cual provoca que los usuarios no puedan acceder concurrentemente a los datos del modelo.

1.6 Conclusiones Parciales

El análisis de las principales técnicas de modelado geológico permitió determinar que la modelación geológica regional utilizando modelos de bloques, es la principal técnica empleada por geólogos y mineros para modelar yacimientos minerales, ya que la misma permite la estimación de los recursos minerales y la planificación de la producción minera. Después analizar los principales sistemas informáticos utilizados en Cuba y en el resto del mundo para la creación y manipulación de modelos de bloques, se evidencia que estos sistemas de carácter privativo son desarrollados por grandes compañías extranjeras y sus precios de venta son muy altos. Teniendo en cuenta lo antes mencionado, la adquisición de estos sistemas por las empresas cubanas que realizan la pequeña minería es casi imposible. La mayoría de estas aplicaciones mencionadas en el presente capítulo, utilizan sistemas gestores de base de datos de escritorio que no permiten la conexión concurrente de múltiples usuarios, lo cual hace engorroso la gestión centralizada de los datos en un ambiente empresarial. Todo esto proporcionó los conocimientos básicos necesarios para continuar con el desarrollo de la investigación.

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS ACTUALES.

El empleo de un conjunto de tecnologías y herramientas informáticas facilita el desarrollo del software aumentando la eficiencia y disminuyendo el coste de este proceso. La investigación tecnológica que incluye herramientas, biblioteca de clases, *frameworks*, estilos, patrones y prácticas, ayuda a alcanzar una arquitectura de trabajo sólida. Para ello se debe realizar un estudio profundo de las herramientas y lenguajes que se adecuan al desarrollo de la aplicación, donde estas presentan un espectro de funciones correspondientes a las necesidades. Para poder desarrollar de forma eficiente un producto informático es necesario hacer un análisis profundo de las tecnologías y sus tendencias existentes en el mundo para así poder definir la óptima. En este capítulo se hace un análisis exhaustivo donde se llegarán a conclusiones acerca de cuáles serán las principales tecnologías y herramientas que se utilizan en la actualidad.

2.1 Lenguajes de Programación.

En este acápite se introducirán los diferentes lenguajes de programación a tener en cuenta para la codificación de la aplicación. Los lenguajes de programación son aquellos sistemas de comunicación que, con una cierta estructura sintáctica y semántica, indica distintas instrucciones a un programa de computadora, permitiendo así crear nuevos programas y software (LOUDEN and LAMBERT, 2011). Existen diversos lenguajes de programación, lo que ha llevado al desarrollo de intérpretes y compiladores. Entre los principales lenguajes están: Visual Studio, Java, C/C++ y C#.

2.1.2 C#

El lenguaje de programación C# es un lenguaje de programación orientado a objetos y a componentes. Es una evolución de los lenguajes C y C++ e incorpora las ventajas o mejoras que tiene el lenguaje JAVA. Algunas de las características del lenguaje de programación C# se fundamentan en su código íntegramente orientado a objeto que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en el Framework .NET. Permite el ahorro de tiempo en la programación, porque tiene una biblioteca de clases muy completa y bien diseñada, posee además características necesarias como la herencia, polimorfismo, encapsulación y los métodos virtuales (TROELSEN, 2010).

Otra de las características básicas de C# son las siguientes:

- Los programas de C# generalmente no tienen necesidad de puntero para la manipulación.
- Forma los constructores sintácticos para las clases, interfaces, estructuras, enumeraciones, entre otros.
- Soporta la programación basada en atributos.

2.1.3 C/C++

El lenguaje de programación C/C++, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C. El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Es un lenguaje híbrido, que se puede compilar y resulta más sencillo de aprender para los programadores que ya conocen C (STROUSTRUP, 2000). El programador tiene el control total de lo que está haciendo, permitiendo una máxima eficiencia al no incorporar verificación de errores en tiempo de ejecución. Las principales características de C/C++ son: el soporte para programación orientada a objetos y el soporte de plantillas o programación genérica. Por ende, se puede decir que C/C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Posee una serie de propiedades por mencionar algunas de ellas: posibilidad de redefinir los operadores (sobrecarga de operadores) e identificación de tipos en tiempo de ejecución, sus siglas en inglés (RTTI)⁴. Está considerado por muchos especialistas como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatización trae lo que dificulta su aprendizaje. Los compiladores de C/C++ generan código nativo con un alto grado de optimización en memoria y velocidad, minimizando la implementación de recursos tales como el polimorfismo y la expansión de patrones lo que lo convierte en uno de los lenguajes más eficientes. La libre utilización de apuntadores por parte de los programadores, por un lado aporta eficiencia, pero por otro es una fuente de errores de lógica. Por este motivo, lenguajes derivados de C/C++, como C# y Java, quitaron este recurso y solo permiten referencias a objetos. Otra fuente de errores es la administración de

⁴ Run time type information: en idioma español, identificación de tipos en tiempo de ejecución.

la memoria ya que la asignación y liberación de memoria dinámica es responsabilidad del programador (DAVIS, 2009).

2.2 Frameworks

Un *framework*⁵ para el desarrollo de software es un esquema para el desarrollo y/o la implementación de una aplicación. Proporcionan un conjunto de bibliotecas de clase con funcionalidades que aumentan la facilidad del trabajo y disminuyen su complejidad. Además permiten reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. De forma general un framework es un conjunto de clases que cooperan y forman un diseño reutilizable formando una infraestructura que facilita y agiliza el desarrollo de aplicaciones (DINUBILE and PATRICK, 2005).

2.2.2 Framework Qt

El framework Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario producida por la división de software Qt de la empresa Nokia⁶, creado por la empresa noruega *Trolltech*. Este sistema de desarrollo es robusto y ha posibilitado la implementación de grandes proyectos como Entorno de Escritorio, *Google Earth*, *Skype* y *VirtualBox*. Distribuida bajo los términos de GNU *Lesser General Public License*, es software libre y de código abierto. Cuenta actualmente con un sistema de triple licencia: GPL v2/v3 para el desarrollo de software de código abierto y software libre y la licencia de pago *Q Public License* (QPL) para el desarrollo de aplicaciones comerciales (THELIN, 2007).

Qt se encuentra programado en forma nativa por el lenguaje C/C++, pero a través de *bindings*⁷ es posible extenderlo para otros lenguajes como *Python*, *Ruby* y *Perl*. Desde hace un tiempo comenzaron a desarrollar sus propias herramientas ideales para los programadores, como es el caso de QtCreator para programadores C/C++, Qt Quick para desarrolladores de interfaces y Qt para dispositivos Nokia. Una de las principales ventajas para los programadores es su extensa documentación disponible, y lo fácil que puede ser internacionalizar una aplicación (SURHONE, LAMBERT *et al.*, 2010b).

⁵ Marco de trabajo.

⁶ Empresa multinacional finlandesa de comunicaciones y tecnología con sede en Keilaniemi, Finlandia.

⁷ Adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita.

2.2.3 Visualization Toolkit (VTK).

El conjunto de herramientas de visualización (VTK) es de código abierto, sistema de software libre disponible para gráficos 3D por computadora, procesamiento de imágenes y la visualización. VTK consiste en un conjunto de bibliotecas de clases de C++ desarrollado por la compañía *Kitware* fundada en 1998 que posee intérpretes para lenguajes script como lo son TCL⁸, Java y Python, lo que aumenta y facilita la posibilidad de implementar aplicaciones en poco tiempo. Consiste en un sistema de visualización por software que permite la representación de geometrías bidimensionales y tridimensionales, soportando una amplia variedad de algoritmos de visualización y modelado. Además se ejecuta bajo distintas plataformas como Windows, GNU/Linux y Macintosh. Entre los principales algoritmos de visualización que ofrece están: escalar, vector, tensor, la textura, los métodos volumétricos, técnicas de modelización avanzadas como modelado implícito, la reducción de polígonos, suavizado de malla, corte, contorno, y la triangulación de Delaunay (AVILA and SOBIERAJSKI, 2012).

La herramienta VTK como es de código abierto, ha impulsado una gran difusión de la misma, extendiendo su aplicación a prácticamente todos los campos en los que se emplean objetos 3D entre los que destacan la medicina, las herramientas industriales, procesamiento paralelo a gran escala, la exploración petrolífera, la mecánica de fluidos y la reconstrucción de superficies a partir de la digitalización con láser. Posee la capacidad de leer una gran cantidad de formatos de imágenes como los son los archivos vtk, obj, bmp, png, entre otros, así como ingresar a la escena tridimensional texto plano, como también texto tridimensional y brinda además varios Widgets⁹ para interfaces de usuarios y controles de cámara como son acercamiento, alejamiento y recorrido de la escena (SCHROEDER *et al.*, 1998).

2.3 Integrated development environment (IDE).

Un IDE es un programa compuesto por un conjunto de herramientas que facilita al programador el desarrollo de un software. Los IDEs por lo general contienen un editor de código, un compilador, un depurador y un constructor de interfaces gráficas. Los IDEs más modernos incluyen un navegador de clases, un inspector de objetos y se integran con sistemas de control de versiones. Los mismos están hechos para soportar uno o más lenguajes de programación (SMITH and SAIVENDY, 2001).

⁸ Tool Command Language: en idioma español, Lenguaje de Herramientas de Comando.

⁹ Componente gráfico o control con el cual el usuario interactúa.

2.3.1 Visual Studio 2010

El IDE Visual Studio 2010 (VS) ha sido el medio principal de desarrollo integrado de software por varios años, dispone de las soluciones, 2010 Profesional, 2010 Premium, 2010 Ultimate y Test Profesional. Incluye un paquete potente de herramientas de administración del ciclo de vida de las aplicaciones para equipos. Con este paquete se puede garantizar la calidad de los resultados, desde el diseño hasta la implementación. VS puede personalizar muchas partes del entorno de trabajo, incluyendo colores, opciones de editor, y el diseño. Este IDE introduce otras características, tal como una jerarquía de llamadas, que le permite ver las rutas de llamadas en el código, fragmentos, que permite escribir una abreviatura que se expande a una plantilla de código, y listas de acciones para generación automática de código nuevo (RANDOLPH *et al.*, 2010).

El editor VS optimiza su experiencia de codificación. Gran parte de su código está coloreado, consejos que aparecen a medida que escribe, y atajos de teclado para realizar una multitud de tareas. Estas características le ayudan a mejorar rápidamente la organización de su código mientras esta codificando. Para una mayor personalización sofisticada, VS expone una interfaz de programación de aplicaciones (API) para la creación de complementos y extensiones. Incorpora herramientas avanzadas para garantizar la calidad del código en todo momento. Es compatible con varias plataformas, entre las que se incluyen Windows, Office, entre otras, todo en un único entorno de desarrollo integrado. Debido a la construcción de capacidades de refactorización, esta se ha convertido en una habilidad principal para los usuarios (MOORE, 2010).

2.3.2 QtCreator

QtCreator es un excelente IDE multiplataforma para desarrollar aplicaciones en C/C++ de manera sencilla y rápida. Además este IDE se ajusta a las necesidades de los desarrolladores Qt. El depurador visual para C/C++ es consciente para la estructura de muchas clases de Qt, lo que aumenta la capacidad de mostrar los datos con claridad. Posee un entorno integrado para la creación y diseño de formas para proyectos C/C++ que permite diseñar rápidamente widgets y diálogos que se usarán en su aplicación. Las formas son totalmente funcionales y se pueden pre visualizar inmediatamente para asegurarse de que se verá y sentirá exactamente como se pensó. Es distribuido bajo tres tipos de licencias: Qt Commercial Developer

License, Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y está disponible para las plataformas: GNU/Linux, Mac OSX, Windows, entre otros ([SURHONE, L et al., 2010a](#)).

El IDE QtCreator como su nombre lo indica, está basado en la biblioteca de clases Qt y cuenta con las siguientes características principales:

- Editor avanzado para C++.
- Diseñador de formularios integrado.
- Herramientas para la administración y construcción de proyectos.
- Completado automático.
- Depurador visual.
- Resaltado y auto-completado del código.
- Ayuda integrada sensible al contexto.
- Soporte para refactorización de código.
- Herramientas para la rápida navegación del código.

2.4 Sistema Gestor de Base de Datos (SGBD).

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando la seguridad e integridad de los mismos ([CABELLO, 2010](#)).

2.4.1 PostgreSQL.

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre. Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como Oracle ([DOUGLAS and DOUGLAS, 2003](#)).

La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x:

- SGBD objeto relacional: Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.
- Altamente extensible: Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Soporte SQL: Soporta la especificación e incluye características avanzadas tales como las uniones.
- Integridad referencial: Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- API flexible: Proporciona soporte al desarrollo fácilmente para el PostgreSQL. Estas interfaces incluyen Python, Perl, PHP, C/C++, entre otros.
- Lenguajes procedurales: Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

2.4.2 SQL Server

SQL Server 2005 es un multi-componente del sistema relacional de gestión de base de datos centrada alrededor de un alto rendimiento. El motor de base de datos en SQL Server es de alta disponibilidad y esto es fundamental para la fiabilidad de los SQL Server 2005 en el manejo de grandes cantidades de

datos. Sin embargo, SQL Server 2005 es mucho más que un motor de base de datos y se compone de un conjunto de herramientas y componentes que se apoyan en el diseño, gestión, mantenimiento y programación. Son muchas las características y herramientas que se encuentran presente en este SGBD por mencionar algunas de las más importantes:

- Empresa: Proporciona una base de datos relacional para satisfacer las necesidades exigentes de las empresas.
- Estándar: Satisface las necesidades de las medianas empresas o departamentos de grandes en las grandes empresas.
- Grupo de trabajo: Satisface las necesidades de las pequeñas y medianas empresas que no requieren de las características de la Edición Estándar.

SQL Server 2005 es una base de datos cliente-servidor. Por lo general, el Servidor SQL 2005 motor de base de datos se instala en un servidor al que se conecta a partir de unas pocas máquinas a cientos o miles de máquinas cliente. Una arquitectura cliente-servidor puede manejar grandes cantidades de datos mejor que un escritorio de base de datos como Microsoft Access. La instancia de SQL Server proporciona seguridad, disponibilidad y fiabilidad, características que están ausentes de las bases de datos tales como Access. Una arquitectura cliente-servidor también puede reducir el tráfico de red. El lado del servidor de una instalación de SQL Server se utiliza para dos grandes categorías de procesamiento de datos: Procesamiento de Transacciones en Línea (OLTP) y Procesamiento Analítico en Línea (OLAP) (DOBSON, 2005).

2.5 Metodologías de Desarrollo.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y de ayudas para la documentación en el desarrollo de productos software. En este se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben tener. Actualmente existen muchas de estas que están caracterizadas en distintas tendencias de desarrollo de software, entre las más conocidas tenemos RUP, AUP y FDD, a continuación se exponen sus principales características.

2.5.1 Rational Unified Process (RUP).

El Proceso Unificado de Desarrollo (RUP) es una infraestructura flexible de desarrollo de software y de arquitectura configurable. Este proceso junto con el Lenguaje Unificado de Modelado, 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. Además este proceso organiza los proyectos en términos de disciplina y fases, consistiendo cada una en una o más iteraciones. Con esta aproximación iterativa el trabajo variará a través del ciclo de vida.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo a necesidades. Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el Proceso Unificado de Desarrollo, que se vendiera como producto independiente (KROLL and KRUCHTEN, 2003).

2.5.2 Agile Unified Process (AUP).

El Proceso Unificado Ágil de Scott Amber es una versión simplificada del Proceso Unificado de Rational (RUP). En él se describe un método sencillo, fácil de entender para el desarrollo de software de aplicaciones empresariales utilizando técnicas ágiles y conceptos que aún permanecen todavía fieles a RUP. Esta metodología abarca siete flujos de trabajo, cuatro de ellos son ingenieriles y los otros tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. Dispone además de cuatro fases igual que RUP: Incepción o Creación, Elaboración, Construcción y Transición. Este enfoque incluye diversas técnicas ágiles, Desarrollo Dirigido por Pruebas, Modelado Ágil, Gestión del Cambio Ágil y Refactorización de Base de Datos para mejorar su productividad (ANDERSON, 2004).

La metodología AUP se basa en los siguientes principios:

- Simplicidad: Todo se describe concisamente utilizando poca documentación, no miles de ellas.

- Agilidad: El ajuste a los valores y principios de *La Alianza Ágil*.
- Centrarse en actividades de alto valor: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
- Herramienta de la independencia: Puede usar cualquier conjunto de herramientas que desea con el AUP.

Algunos de los principios contenidos en *La Alianza Ágil* (PRIOLO, 2009).

- La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

2.6 Herramientas CASE.

Las herramientas CASE¹⁰ 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 dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

¹⁰ Siglas de Computer Aided Software Engineering, en el idioma español: Ingeniería de Software Asistida por Ordenador.

2.6.1 Visual Paradigm.

Visual Paradigm para UML es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite crear diferentes tipos de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta de modelado también proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos ([ZIELIŃSKI and SZMUC, 2006](#)).

Dentro de sus principales características se mencionan algunas de ellas como:

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

2.6.2 Rational Rose.

Rational Rose es una herramienta potente de modelado visual para el análisis y diseño de sistemas orientado a objetos. Se utiliza para modelar el sistema antes de escribir algún código. Esta herramienta soporta el modelo de negocio, que ayuda a un mejor entendimiento del negocio en todo el sistema. Además permite el análisis de sistemas, ya que permite el diseño de casos de uso y diagramas de Casos de Uso para mostrar el sistema funcionalidad. También se puede diseñar diagramas de interacción, para

mostrar cómo los objetos trabajan juntos para proporcionar la funcionalidad necesaria. Los diagramas de clases pueden ser creados para mostrar las clases de un sistema y cómo se relacionan entre sí. Los diagramas de componentes pueden ser desarrollados para ilustrar cómo las clases se asignan a la aplicación componentes. Por último, un diagrama de implementación puede ser producido para mostrar el diseño de la red para el sistema ([BOGGS and BOGGS, 2002](#)).

La herramienta Rational soporta ocho tipos diferentes de diagramas UML: diagramas de casos de uso, diagramas de actividad, diagramas de secuencia, diagramas de colaboración, los diagramas de clases, diagramas de estados, diagramas de componentes y diagramas de despliegue. Utiliza un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación. Algunas de las características que define esta herramienta son las siguientes:

- Genera documentación automáticamente.
- Generación de código a partir de los modelos.
- Ingeniería inversa, es decir se crean modelos a partir del código.
- Mantiene la consistencia de los modelos del sistema software.

2.7 Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar ([LÓPEZ et al., 2004](#)).

UML proporciona una forma estándar de escribir los planos de un sistema, tales como procesos del negocio, funciones del sistema, clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables. Es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requisitos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código (BOOCH *et al.*, 2005).

2.8 Herramientas y tecnologías seleccionadas para la investigación.

Para el desarrollo de la aplicación se escogen las siguientes herramientas y tecnologías definidas en la **Tabla 1**, las cuales son compatibles con las herramientas que se utilizan en el proyecto Sistema Minero Cubano para el desarrollo de sus productos. Algunas herramientas y tecnologías seleccionadas para el desarrollo de la solución propuesta están justificadas en el documento Vista Entorno de Desarrollo Tecnológico (*0120_Arquitectura de Software.doc*), elaborado por el Arquitecto de Software del proyecto y el mismo se encuentra en el Expediente del Proyecto.

Herramientas y Tecnologías escogidas		Versión	Utilizada en el proyecto SMC
Lenguaje de programación	C/C++	4.6	Sí
Frameworks	Qt framework	4.8	Sí
	Visualization Tool Kit (VTK)	5.8	Sí
Sistema Gestor de Base de Datos	PostgreSQL	9.1	Sí
Extensión espacial para Postgresql	Postgis	1.5.3	Sí
Entorno Integrado de Desarrollo	QtCreator	2.5	Sí

Metodología de Desarrollo	AUP		No
Lenguaje de modelado	UML	2.0	Sí
Herramienta de modelado	Visual Paradigm for UML	8.0	Sí

Tabla 1: Herramientas y Tecnologías escogidas.

Las herramientas y tecnologías mencionadas anteriormente en la **Tabla 1**, en su mayoría son de código abierto y multiplataforma. Además se seleccionó el lenguaje de programación C/C++, el mismo es el lenguaje nativo utilizado para el trabajo con gráficos por computadora. Con el objetivo de disminuir el tiempo de desarrollo se seleccionó la biblioteca de clases Visualization Toolkit (VTK), la cual consta con la implementación de miles de algoritmos de geometría computacional. La metodología de desarrollo de software Agile Unified Process (AUP), la cual está basada en la metodología Rational Unified Process (RUP) permite el desarrollo ágil de proyectos informáticos, en el caso particular de la presente investigación, AUP se adecua al desarrollo de la solución propuesta.

2.9 Conclusiones Parciales.

En este capítulo se hace un análisis de las principales herramientas y tecnologías necesarias para desarrollar la aplicación. Las herramientas y tecnologías escogidas son de código abierto y compatible con las tecnologías utilizadas en el proyecto Sistema Minero Cubano (SMC), esta selección permitirá que la propuesta de solución sea fácil de integrar con los productos desarrollados por el proyecto SMC. Las tecnologías y herramientas seleccionadas para el desarrollo de la propuesta de solución garantizan que la misma sea una herramienta informática multiplataforma, cumpliendo así con las políticas de migración al software libre definidas en el país. En este capítulo se seleccionó el lenguaje de programación C/C++ y la biblioteca de clases VTK lo cual permitirá que la propuesta de solución tenga una robusta implementación del ambiente gráfico tridimensional.

CAPÍTULO 3: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

Para el desarrollo exitoso de la aplicación a desarrollar es necesario tener un amplio conocimiento de los procesos que ocurren en la actualidad en la administración y gestión de la explotación de yacimientos minerales, así como identificar los principales requerimientos funcionales y no funcionales que debe tener el sistema para posteriormente detallar sus principales procesos. Aspectos como los antes mencionados se comprenden mejor modelándolos mediante el conjunto de normativas y artefactos que proporciona el RUP; por todo esto en el presente capítulo se realizará el modelo del dominio, se identificarán y establecerán el conjunto de requisitos tanto funcionales como no funcionales existentes y se modelará el sistema completamente, todos estos con sus artefactos correspondientes.

3.1 Modelo de Dominio.

El modelo de dominio se realiza con el objetivo de comprender y describir las clases más importantes dentro del contexto del sistema y, por tanto, también contribuir a la comprensión de los requisitos de dicho sistema que se desprenden de este contexto (SOMMERVILLE, 2005). Su descripción se realiza mediante diagramas UML (específicamente mediante diagramas de clases), ver **Figura 7**.

3.1.2 Diagramas de clases del dominio.

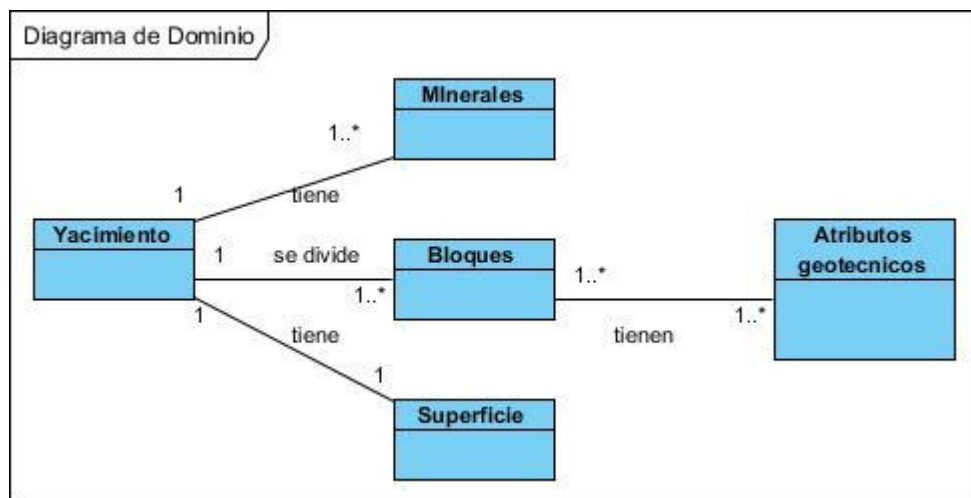


Figura 7: Diagrama de dominio.

3.1.3 Descripción del modelo de dominio.

En la modelación geológica tridimensional un yacimiento mineral puede ser modelado utilizando la técnica de modelado llamada Modelo de Bloques. Esta técnica consiste en dividir un yacimiento mineral en bloques que son celdas tridimensionales en forma de paralelepípedos. Cada bloque tiene uno o más atributos geotécnicos en los cuales se registra sus valores, como por ejemplo el tipo de roca y el valor económico del mismo. Además un yacimiento tiene una superficie topográfica y a la vez contiene diversos minerales.

3.2 Requisitos funcionales y no funcionales del sistema.

En este acápite se muestran en forma de listado los distintos requisitos funcionales y no funcionales del sistema. Los requisitos funcionales son condiciones o capacidades que el sistema debe cumplir y los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (PRESSMAN, 2010). A continuación se listan los requisitos funcionales y no funcionales que el sistema debe tener, teniendo como prefijo los requisitos funcionales las letras RF y de los no funcionales las letras RNF:

Requisitos funcionales:

- RF 1. Crear nuevo proyecto.
- RF 2. Abrir proyecto existente.
- RF 3. Listar base de datos existentes.
- RF 4. Establecer conexión a la base de datos.
- RF 5. Crear estructura básica de las tablas del modelo de bloques en la base de datos.
- RF 6. Salvar los datos del proyecto en un fichero de configuración.
- RF 7. Listar los modelos de bloques existentes en la base de datos.
- RF 8. Listar los bloques dado un modelo de bloques.

- RF 9. Mostrar las propiedades de un modelo de bloques.
- RF 10. Crear un modelo de bloques.
- RF 11. Eliminar un modelo de bloques.
- RF 12. Calcular el nivel básico de resolución.
- RF 13. Aplicar transformaciones a un modelo de bloques.
- RF 14. Adicionar atributos a los bloques del modelo de bloque.
- RF 15. Adicionar bloques al modelado.
- RF 16. Editar los atributos de un bloque del modelo.
- RF 17. Eliminar bloques del modelo.
- RF 18. Importar un modelo de bloques desde un fichero de texto.
- RF 19. Buscar bloques por rango de localización.
- RF 20. Visualizar tridimensionalmente los bloques del modelo.
- RF 21. Limpiar escena.
- RF 22. Cambiar color del fondo de la escena.
- RF 23. Editar las propiedades visuales del modelo de bloques.

Requisitos no funcionales:

- **Requisitos de Usabilidad:**

RNF 1: El sistema debe poder ser usado por cualquier persona que tenga conocimientos básicos de computación.

RNF 2: La información deberá estar disponible en todo momento, limitada solamente por las restricciones que ésta tenga de acuerdo a las políticas de seguridad del sistema.

RNF 3: Se debe lograr un producto altamente configurable y extensible, de manera que sea posible incorporar a éste nuevas funcionalidades, sin que se afecte el mismo.

- **Requisitos de Confiabilidad:**

RNF 5: Una vez que ocurra una excepción producto a un error leve el sistema debe informar el mismo y seguir funcionando, si por el contrario es debido a un error crítico, debe informar el error y a continuación cerrarse.

RNF 6: Debe mantenerse la consistencia de los datos en correspondencia con la realidad.

- **Requisitos de Eficiencia:**

RNF 7: El servidor de base de datos debe permitir que se conecten a trabajar concurrentemente hasta 50 usuarios como máximo.

- **Requisitos de Soporte:**

RNF 8: El sistema debe estar codificado de acuerdo al estándar de codificación para C++ establecido en el proyecto y estándares de Base de Datos de SQL.

- **Requisitos de Ayuda y Documentación:**

RNF 9: El sistema debe estar provisto de un manual de usuario, donde se detallen cada uno de los elementos del mismo. Esto permitirá a los usuarios conocer cada una de las funcionalidades que brinda el sistema y como utilizarlas.

- **Requisitos de Interfaz de Usuario:**

RNF 10: El módulo debe tener una interfaz gráfica uniforme incluyendo pantallas, menús y opciones. Las pautas de diseño serán definidas por el comité de rol de desarrolladores de interfaz de usuario.

RNF 11: La consistencia de la interacción entre usuario y componente estará determinada por el diseño de la interfaces de usuario que mantendrán los elementos como menús y zonas de trabajo, en posiciones fijas, además de la mayor uniformidad posible entre cuadros de texto y botones.

RNF 12: Tanto los títulos de los componentes de la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma español y tener una apariencia uniforme. Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error.

RNF 13: Se deberá facilitar la entrada de datos a los usuarios, presentando campos de selección que permitan escoger los valores posibles con los que se podrá llenar un determinado elemento en la interfaz, haciendo que el proceso de llenado de datos sea lo más intuitivo posible, de tal forma que los usuarios se puedan adaptar fácilmente.

- **Requisitos de Software:**

RNF 14: Como sistema gestor de base de datos se hará uso de PostgreSQL, por defecto a través del puerto 5432. Además puede configurarse para soporte conexión segura por el protocolo SSL a través del puerto 443.

RNF 15: Como Sistema Operativo Linux o Windows, recomendándose utilizar Ubuntu 12.0 o superior o Windows (NT o superior).

- **Requisitos de Hardware:**

RNF 16: Las computadoras que utilizarán el software a desarrollar deberán tener 512 MB de Memoria RAM como mínimo.

RNF 17: Se requiere como mínimo un servidor de base de datos de 2 GB de memoria RAM y un procesador INTEL PENTIUM IV o superior, de 2.8 GHz, soporte para 1 núcleo o múltiples núcleos así como un espacio disponible en el disco duro mayor que 8 GB.

3.3 Descripción del sistema propuesto.

El sistema propuesto será una aplicación informática de escritorio, la cual gestionará los datos asociados a un yacimiento mineral a través de un modelo de bloques. Los datos del sistema serán almacenados en una base de datos utilizando el SGBD PostgreSQL y su extensión Postgis para el manejo de los datos

espaciales del modelo. El sistema permitirá la visualización tridimensional del modelo de bloques así como la gestión del mismo.

3.3.1 Descripción de los actores.

Los actores del sistema son abstracciones de las entidades externas a este, son subsistemas o clases que interactúan directamente con el sistema. Un actor participa en un caso de uso o en conjunto coherente de casos de uso para llevar a cabo un propósito global. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. A continuación se muestra la **Tabla 2** con los actores del sistema, así como la descripción de cada uno de ellos.

Actor	Descripción
Usuario	Puede ser un geólogo, minero o cualquier otro especialista asociado a la rama, con los que el sistema interactúa dentro de las entidades geológicas mineras o fuera de ellas, que consulte información o interactúe con los modelos de bloques.

Tabla 2: Actores del sistema.

3.3.2 Casos de Uso del Sistema.

A continuación se muestra el Diagrama de Caso de Uso del Sistema (ver **Figura 8**) donde están representados el actor que interactúa con el sistema mediante los casos de uso del sistema y luego la descripción textual de cada uno de estos.

3.3.3 Diagrama de Caso de Uso del Sistema.

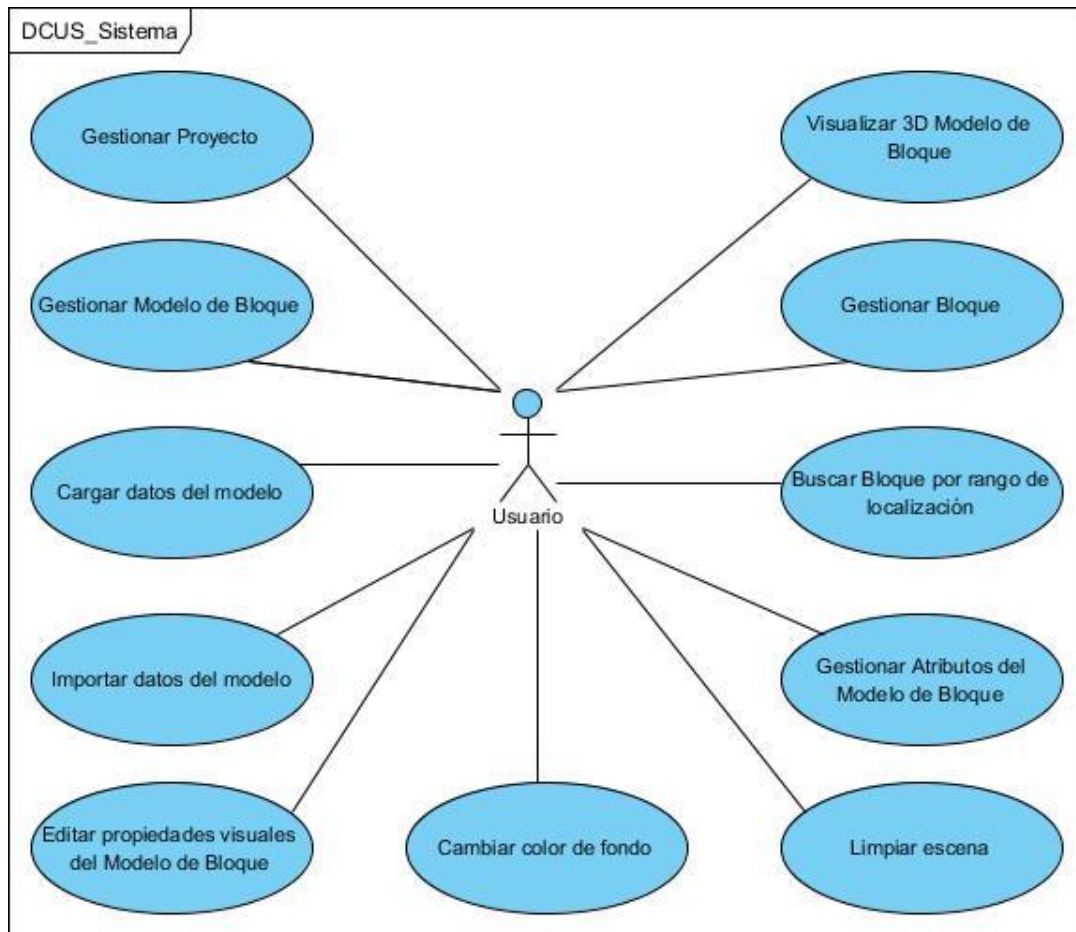


Figura 8: Diagrama de Caso de Uso del Sistema.

3.3.4 Descripción textual de los Casos de Uso del Sistema.

La descripción de los restantes casos de uso se encuentra en el [Anexo 1](#).

Tabla 3: Descripción CUS Gestionar Proyecto.

Caso de Uso:	Gestionar Proyecto
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de poder crear o abrir un proyecto.

Resumen:	El caso de uso se inicializa cuando el usuario desea crear o abrir un proyecto y termina ejecutándose la acción deseada.	
Precondiciones:	Debe estar inicializada la aplicación. La Base de Datos existente tiene que tener la estructura del proyecto.	
Referencias	RF 1, RF 3, RF 4, RF 5, RF 6, RF 7, RF 8	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El caso de uso se inicializa cuando el usuario selecciona las siguientes opciones del menú Archivos:</p> <ul style="list-style-type: none"> - Nuevo proyecto. - Abrir proyecto. 	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> - Si selecciona “Nuevo Proyecto”, ver sección “Nuevo proyecto”. - Si selecciona “Abrir Proyecto”, ver sección “Abrir proyecto”. 	
	<p>3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario y actualiza el árbol de proyecto.</p>	
Sección ‘Nuevo proyecto ’		
Acción del Actor	Respuesta del Sistema	
<p>1. El caso de uso se inicializa cuando el usuario selecciona la opción Nuevo Proyecto del menú Archivos o el icono de la barra de herramientas. Ver Interfaz 1a y 1b.</p>	<p>2. El sistema muestra un formulario Nuevo Proyecto. Ver Interfaz 1c.</p>	
<p>3. El usuario introduce los datos correspondientes a los campos del formulario. Luego presiona el botón Aceptar.</p>	<p>4. El sistema verifica los parámetros, si los parámetros están correctos se muestra un mensaje de conectado y se deshabilita la opción de Base de Datos Existente ir a la sección de</p>	

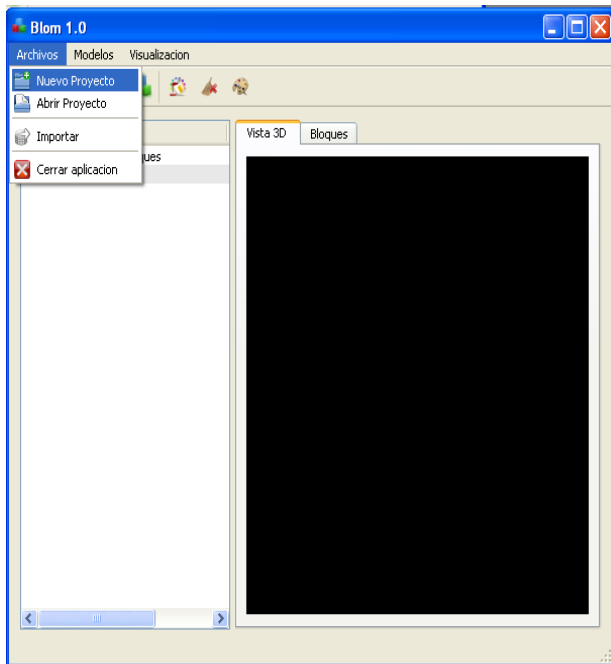
“Base de Datos Existente”.

5. Sistema ejecuta un fichero de texto con las sentencias SQL necesarias para la creación de la base de datos en el Sistema Gestor de Base de Datos del sistema.

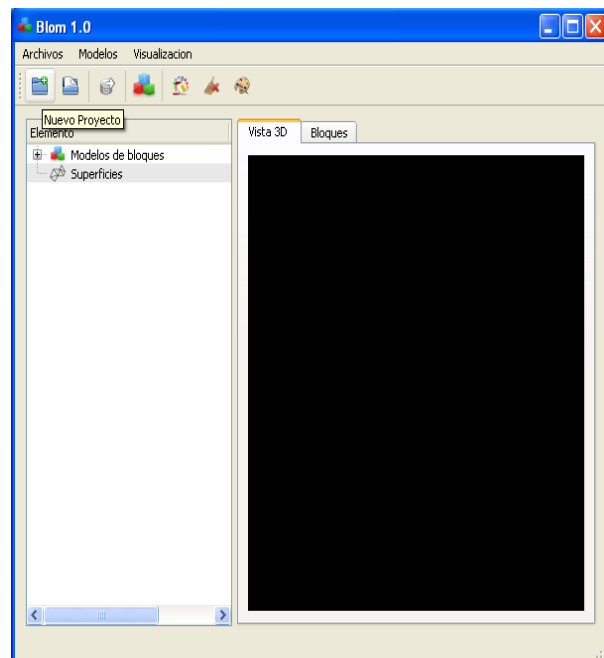
6. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario y se crea el nuevo proyecto.

Prototipo de Interfaz

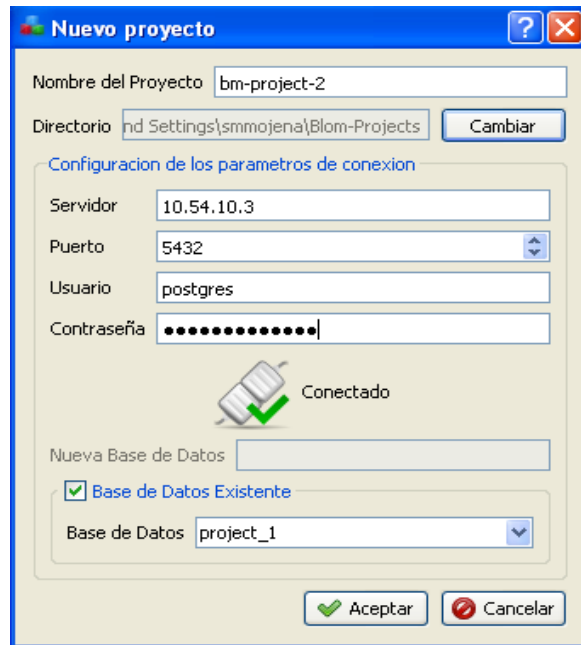
Interfaz 1a



Interfaz 1b



Interfaz 1c



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4. El sistema verifica que los valores entrados son incorrectos y muestra los campos en color rosado, para completar los mismos.

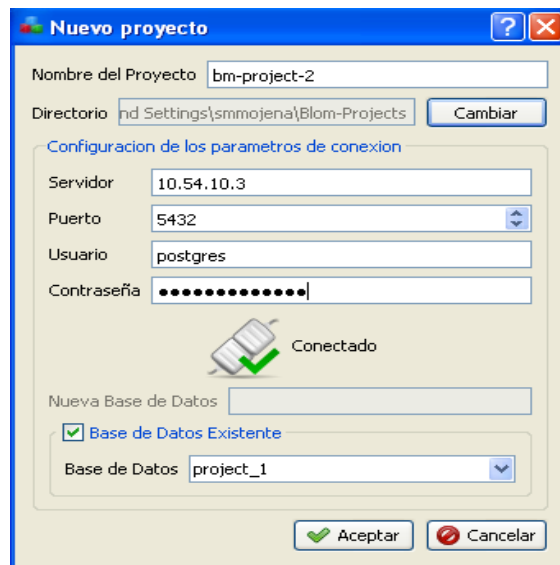
Sección "Base de Datos Existente "

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Base de Datos Existente y selecciona la base de datos que desea. Luego presiona el botón Aceptar. Ver Interfaz 1d.	2. El sistema realiza la operación de Base de Datos Existente seleccionada por el usuario y muestra el proyecto existente.
	3. El caso de uso termina cuando el sistema procesa la información a partir de la acción

realizada por el usuario y visualiza el proyecto.

Prototipo de Interfaz

Interfaz 1d



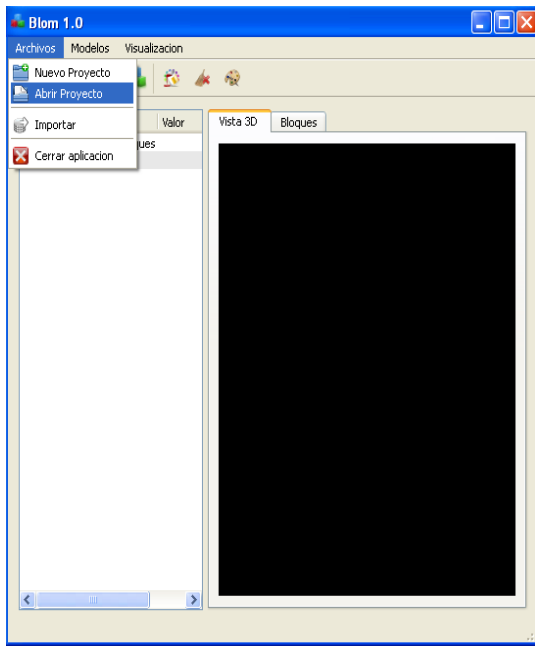
Sección "Abrir proyecto"

Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicializa cuando el usuario selecciona del menú Archivo la opción Abrir Proyecto. Ver Interfaz 2a.</p>	<p>2. El sistema muestra una ventana de diálogo Abrir Proyecto para que el usuario busque en el directorio del sistema operativo el fichero del proyecto. Ver Interfaz 2b.</p>
<p>3. El usuario selecciona el fichero de configuración del proyecto y presiona el botón Aceptar.</p>	<p>4. El sistema lee los datos de configuración del proyecto y verifica si se puede conectar a la base de datos.</p> <p>- Si se establece la conexión, se carga el árbol del proyecto y se carga un modelo de bloques. Ver caso de uso "Cargar Modelo de Bloques".</p> <p>5. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario y actualiza la</p>

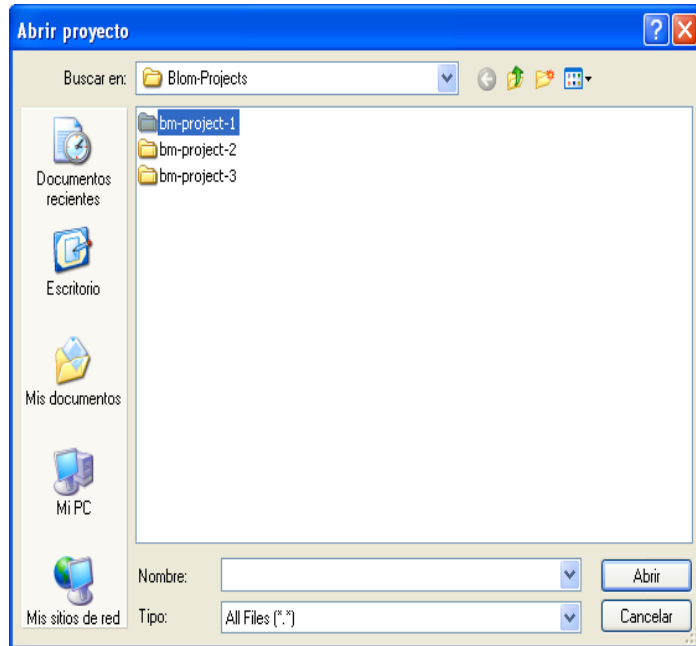
visualización del proyecto con sus diferentes pestañas (vista 3D y bloques).

Prototipo de Interfaz

Interfaz 2a



Interfaz 2b



Flujos Alternos

4. El sistema no establece conexión, se muestra un mensaje de alerta al usuario y no se carga el proyecto.

Poscondiciones

Mostrar la ventana principal del proyecto.
 Se carga en el árbol del proyecto los modelos existentes.
 El sistema crea y visualiza el proyecto a partir de la acción realizada por el usuario.

3.4 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 cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación.

3.4.1 Diagrama de clases del diseño.

El diagrama de clases expresa la estructura u organización del software en términos de las clases. Es un reflejo abstracto de los componentes y las relaciones entre ellos. Los diagramas de clase 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. A continuación se muestra el diagrama de clases del diseño que representa la estructura del sistema, teniendo en cuenta las entidades, sus atributos y relaciones. Los restantes diagramas se encuentran en el [Anexo 2](#).

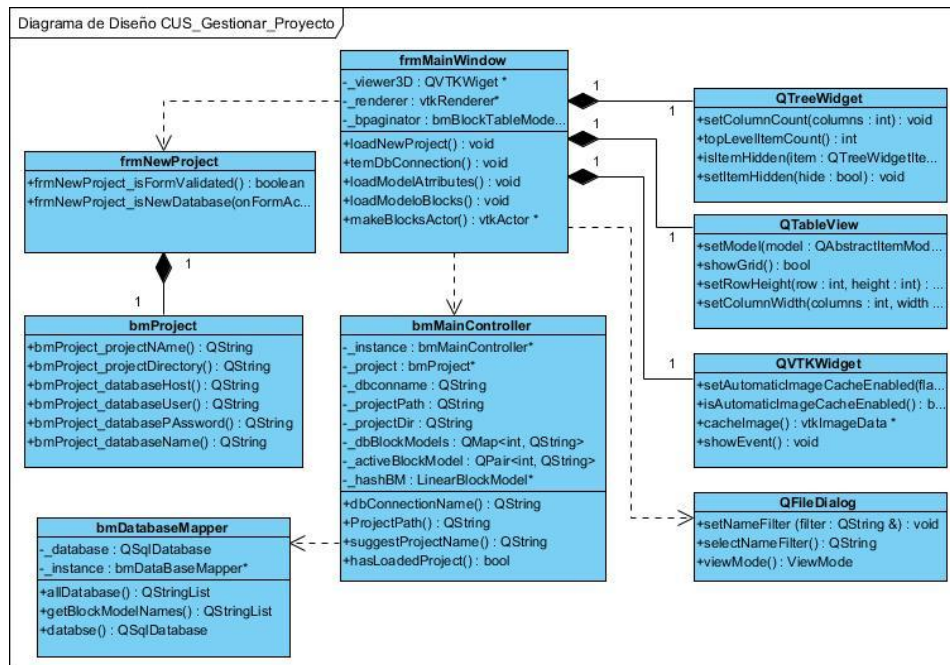


Figura 9: Diagrama de Diseño del Caso de Uso Gestionar Proyecto.

3.5 Diseño de la Base de Datos.

La base de datos necesita de una definición de su estructura, de manera que permita almacenar datos, reconocer el contenido, y recuperar la información. Para diseñar una base de datos es necesario seguir un conjunto de pasos que comienzan con definir las clases persistentes, luego refinarlas y clasificarlas junto con sus atributos, para más tarde realizar el diagrama de clases persistentes.

3.5.1 Diagrama Entidad Relación.

A continuación en la **Figura 10**, se muestra el modelo lógico de la base de datos en el diagrama Entidad-Relación.

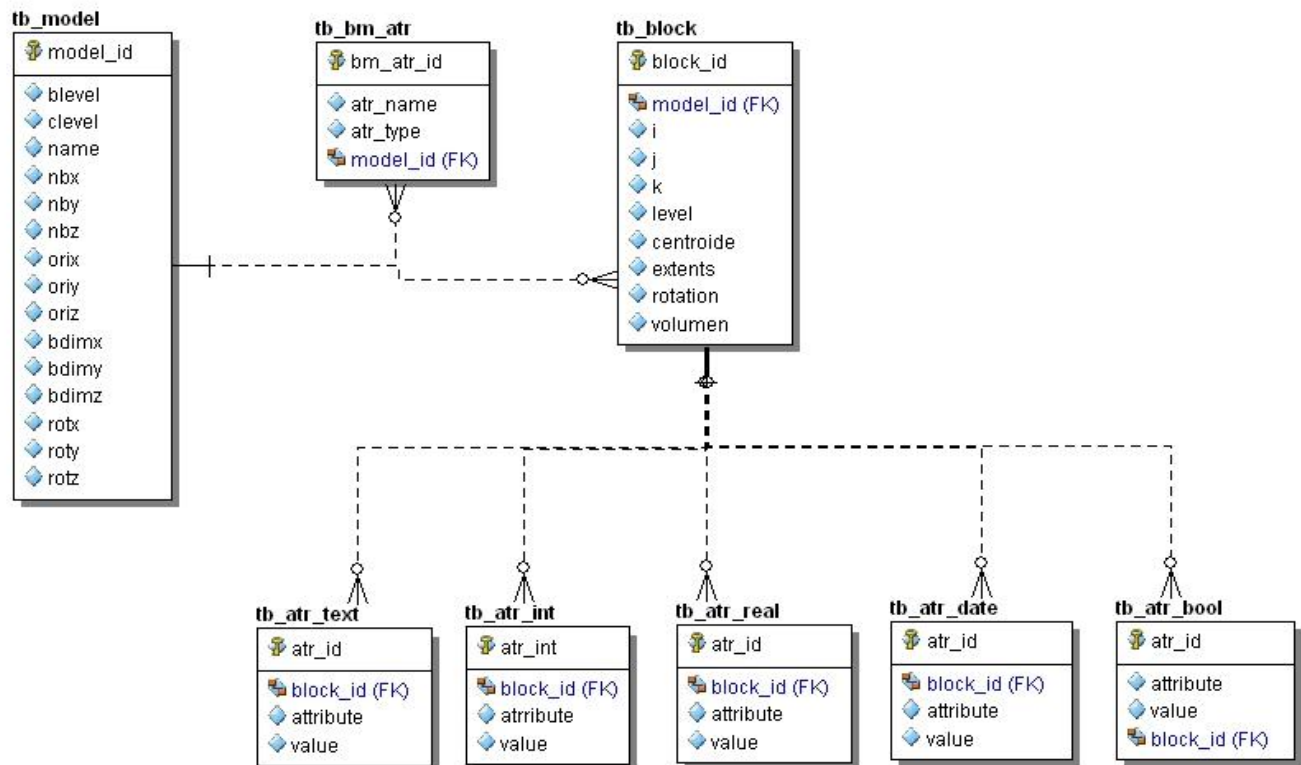


Figura 10: Modelo Entidad Relación.

La base de datos relacional se normaliza para evitar la redundancia de los datos, problemas de actualización de los datos en las tablas y se protege la integridad de los mismos.

3.6 Arquitectura del Sistema.

La arquitectura de software es un elemento fundamental en la construcción de cualquier sistema informático. Permite la comunicación entre las partes involucradas en el desarrollo de un sistema. Además, incluye decisiones fundamentales relativas al diseño de aplicaciones, las cuales impactan el trabajo de ingeniería de software e influyen en el éxito final. Un componente de software se define como un objeto de software diseñado para cumplir con determinado propósito. Los componentes se caracterizan por ser reutilizables, de ahí que puedan ser utilizados por diferentes aplicaciones (PRESSMAN, 2010).

El componente desarrollado emplea la arquitectura basada en componentes. El mismo permite acceder y manipular la información contenida en una BD específica. Además, permite acceder a sus funcionalidades de una forma sencilla y transparente, a través de una interfaz bien definida. Dicha interfaz expone los métodos necesarios para establecer la conexión a una BD, así como ejecutar consultas y transacciones. Otro componente que se emplea es la arquitectura Cliente – Servidor donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

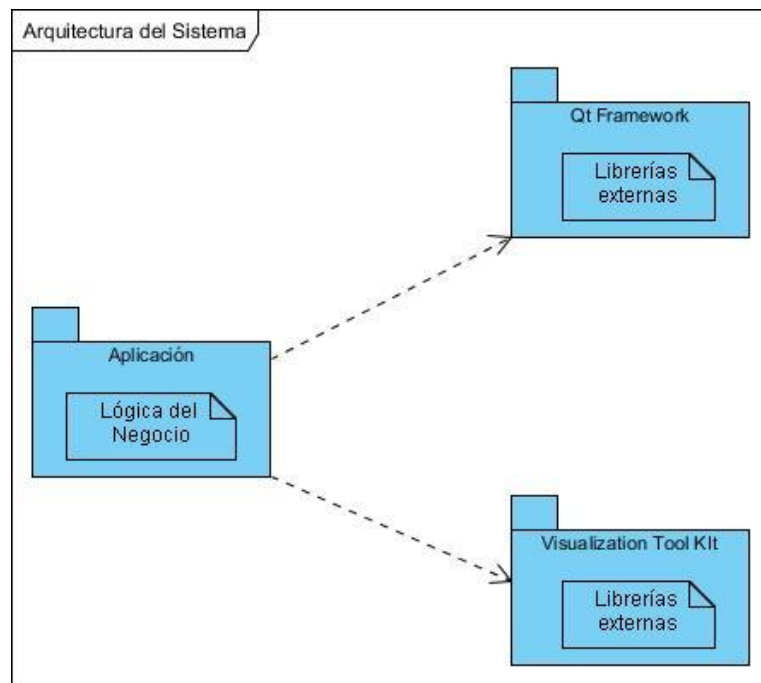


Tabla 4: Diagrama de Arquitectura del Sistema.

3.7 Patrones de Diseño.

Los patrones de diseño a utilizar, tienen como objetivos principales el garantizar una mejor comprensión de la arquitectura del sistema y agilizar el posterior desarrollo e implementación del mismo. A continuación se listan los patrones de diseño a objetos utilizados publicados en el libro "Design Patterns", se conocen comúnmente como GOF (gang of four, "pandilla de los cuatro") (LARMAN, 2004).

Singleton (Instancia única): Garantiza que una clase sólo tiene una única instancia, proporcionando un punto de acceso global a la misma. Para ello en lugar de tener una variable global, la instancia se almacena un atributo estático de la clase y se accede a ella por el método Instance.

Modelo Vista Controlador: es un patrón de diseño de software.

- **Modelo:** Esta es la representación específica del dominio de la información sobre la cual funciona la aplicación.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Observador: Brinda un mecanismo que permite a un componente transmitir de forma flexible mensajes a aquellos objetos que hayan expresado interés en él. Define una dependencia uno a muchos entre objetos, de modo que cuando el estado de un objeto cambia, se les notifica el cambio a todos los que dependen de él y se actualizan de forma automática.

Facade (Fachada): Ofrece un punto de acceso al resto de las clases, si estas cambian o se sustituyen por otras solo hay que actualizar la clase Facade sin que el cambio afecte a las aplicaciones cliente. Este patrón no oculta las clases sino que ofrece una forma más sencilla de acceder a ellas, en los casos en que se requiere se puede acceder directamente a ellas.

3.8 Conclusiones Parciales.

En este capítulo se identificaron los requerimientos funcionales y no funcionales de la solución propuesta, lo cual permitió establecer las pautas para la futura implementación del sistema. También se definió el diseño de la base de datos del sistema teniendo en cuenta las características del SGDB PostgreSQL y de su extensión Postgis lo cual logró la persistencia de los datos de los modelos de bloques y permitió una mayor eficiencia en el manejo de los datos espaciales de los mismos. La arquitectura orientada a componentes definida en este capítulo ofreció un mayor nivel de robustez a la solución propuesta y la utilización de los patrones de diseños mencionados en este capítulo garantizaron las buenas prácticas en la implementación de la solución propuesta.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.

Como resultado de la descripción de la solución del sistema, se procede a la implementación del sistema en términos de componentes: ejecutables y código fuente. Para lograr una correcta implementación, es necesario que la solución propuesta satisfaga las expectativas del usuario final, asegurando que sea operacional o que funcione de acuerdo a los requerimientos. Derivándose entonces la incorporación de pruebas al sistema, actividad para garantizar la calidad del software. En el presente capítulo se exponen, además de los aspectos mencionados anteriormente, aquellos relacionados con los estándares de codificación que permiten la creación de un código fácil de entender y el tratamiento de errores.

4.1 Implementación.

Partiendo de los resultados hasta el momento, con el propósito de clarificar los requisitos restantes y completar el desarrollo del sistema basándose en la línea base de la arquitectura, se procede a la implementación del sistema. Esta fase se puede ver como un proceso de fabricación, en el que se pone énfasis a la gestión de los recursos y el control de las operaciones para optimizar los costes, la planificación y la calidad del producto final.

4.1.1 Estándares de codificación.

Uno de los instrumentos que facilitan la tarea de asegurar la calidad del software es la adopción de estilos y estándares de codificación. El uso de estos estándares tiene innumerables ventajas entre ellas lograr un estilo de código homogéneo asegurando su legibilidad y proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado. Además ayuda a mejorar el proceso de codificación haciéndolo en gran medida eficiente y en muchos casos reutilizables.

Es debido a estos argumentos que la codificación de los módulos del Sistema para la gestión de modelos de bloques debe cumplir con ciertos requisitos, algunos de los cuales son detallados seguidamente:

- Los nombres de las variables y de las funciones deben seguir la notación lowerCamelCase con el prefijo bm para las clases y frm para los formularios.

- Las variables deberán ser siempre tan elocuentes como prácticas.

4.1.2 Tratamiento de errores.

Con el objetivo de garantizar la mayor confiabilidad e integridad en los datos que utiliza el sistema, se adoptan las siguientes estrategias para el tratamiento de errores:

- Se validan los campos de los formularios de las interfaces pertenecientes a los módulos del Sistema para la gestión de modelo de bloques.
- La información que contiene los mensajes de errores detectados se muestra con un lenguaje claro, legible y sencillo de entender.
- Dado el caso que la información introducida en un formulario sea incorrecta o incompleta, se señalarán los campos que contienen el error señalados en color rosado.

Imágenes de ejemplo de tratamiento de errores

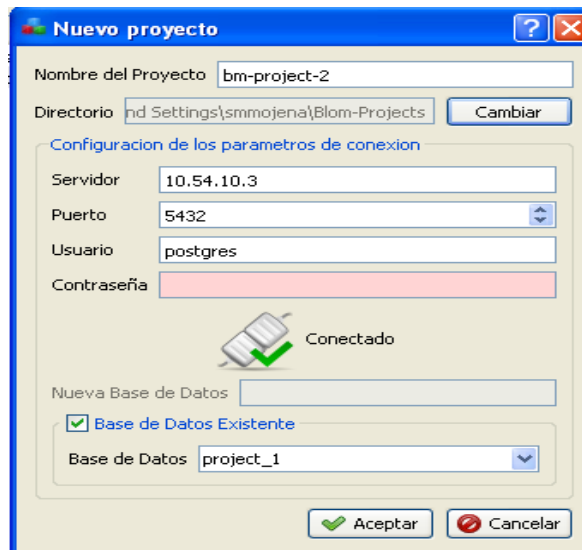


Figura 11: Ejemplo de tratamiento de errores a nivel de interfaz.

4.1.3 Modelo de implementación.

El modelo de implementación describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. Un diagrama de componentes muestra las organizaciones y las dependencias lógicas de componentes de software. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia, pudiendo mostrarse las interfaces que estos soporten. Además muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. El diagrama de componentes de Sistema para la gestión de modelos de bloques (**Ver Figura 2**) se corresponde con la arquitectura definida que da soporte a la aplicación. En él se detallan cómo el sistema está desglosado en componentes y las dependencias entre ellos.

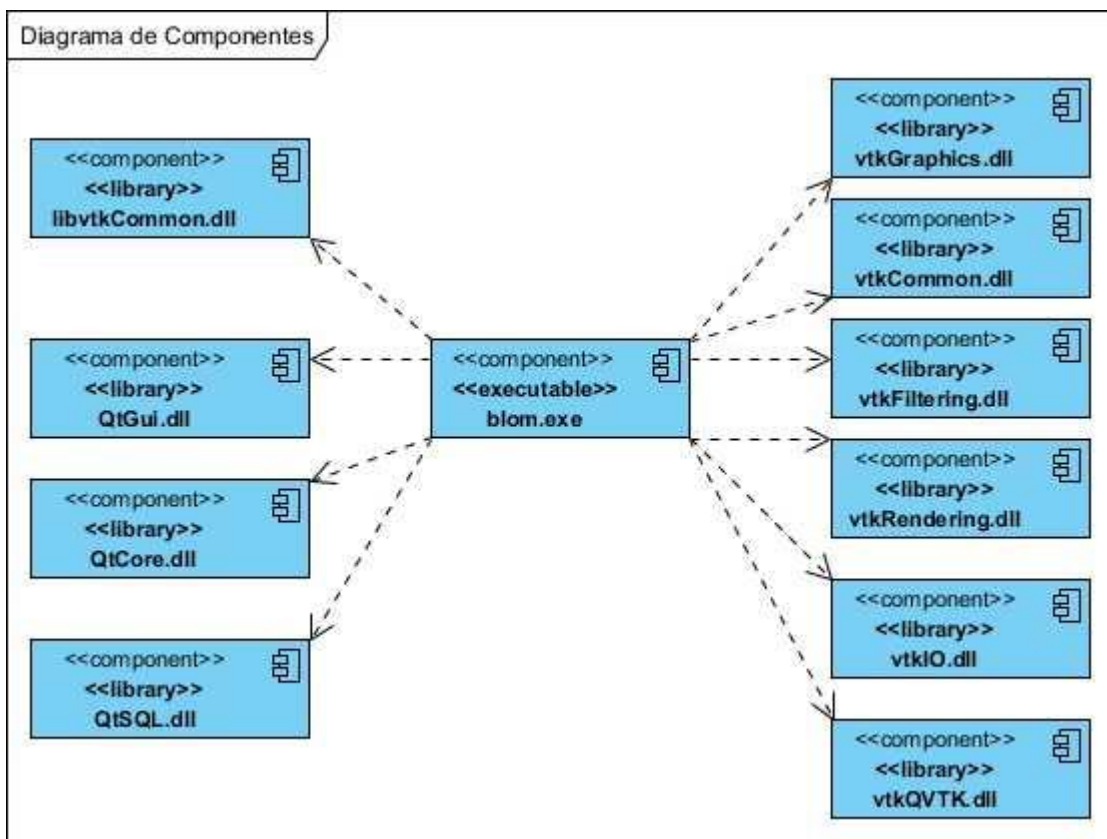


Figura 12: Diagrama de componentes.

4.1.4 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, ver **Figura 123**.

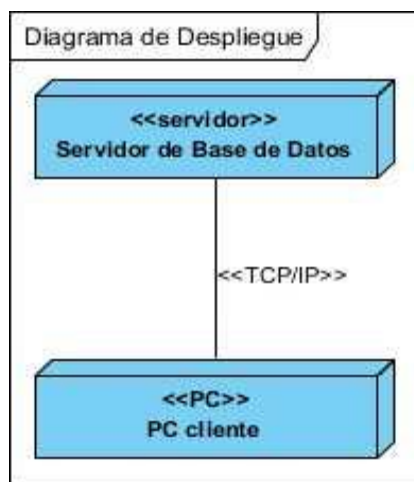


Figura 12: Diagrama de despliegue.

4.2 Prueba 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 requerimientos 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 (PRESSMAN, 2010).

4.2.1 Prueba del sistema.

Las técnicas de caja negra o funcionales son las que se realizan sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del

programa, únicamente la funcionalidad que debe realizar. También conocidas como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba.

Para aplicar las pruebas al sistema se determinaron los casos de pruebas en un conjunto de entradas, condiciones de ejecución y resultados esperados, con tal de verificar el cumplimiento de un objetivo en particular. En la Tabla se muestra el diseño del caso de prueba de la funcionalidad Nuevo Modelo de Bloques, el resto de los casos de pruebas se encuentran en el [Anexo 3](#).

Condiciones de ejecución

1. Debe estar inicializada la aplicación.
2. El usuario tiene que seleccionar la opción Crear Proyecto.

SC Nuevo Modelo de Bloques

Escenario	Variable 1	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Crear Modelo de Bloques	V	Permite crear un modelo de bloques.	Muestra un formulario Nuevo Modelo de Bloques.	Paso 1: Selecciona la opción Nuevo Modelo de Bloques del menú Modelos. Paso 2: Introduce los datos correspondientes a los campos de la ventana y presiona el botón aceptar. Paso 3: Continúa en el paso 2 del flujo básico "Nuevo Modelo de Bloques".
	text			
	V spin box			

Figura 13: Diseño del caso de prueba Nuevo Modelo de Bloques.

Los resultados obtenidos después de aplicar las pruebas de sistema son las siguientes:

Iteración	No conformidades
1	6
2	3
3	0

Tabla 5: Iteraciones de prueba de caja negra

En la **Tabla 5**, se comienza por la primera iteración con 6 no conformidades de 23 requisitos funcionales. Luego se prosigue a una segunda iteración disminuyendo en esta a 3 no conformidades. Posteriormente se ejecuta la tercera y última iteración donde se eliminan todas las no conformidades, obteniendo un buen funcionamiento de la herramienta que gestiona los datos asociados a un modelo de bloques.

4.3 Caso de estudio.

Con el objetivo de comprobar que las funcionalidades del sistema propuesto cumplen con los objetivos planteados en la investigación, se describe un caso de estudio en el cual se utilizará la sección O48 de un yacimiento de níquel cubano. Esta sección del yacimiento es utilizada por los científicos cubanos de las ramas de las Geociencias para realizar diferentes investigaciones. En este experimento se creará un modelo de bloques, se le importarán los datos de otro modelo de bloques similar hecho en el sistema Gemcom-Gems y se visualizará de varias formas los datos del modelo. A continuación se muestra en la **Tabla 6** los datos básicos del modelo de bloques que se creará.

Coordenadas del origen	(4733.33, 5533.34, 69)
Dimensiones del modelo	16 x 16 x 22
Dimensiones de cada bloques del modelo	16.66 x 16.66 x 3
Vector de rotación	(0, 0, 0)
Total de bloques del modelo	5632

Tabla 6: Datos básicos del modelo de bloques.

Paso 1: Abrir la herramienta y crear un nuevo proyecto.

En este paso se estableció la conexión al servidor de bases de datos y se creó una nueva base de datos con el nombre “*db_bm_o48*” como se muestra en la **Figura 14**.

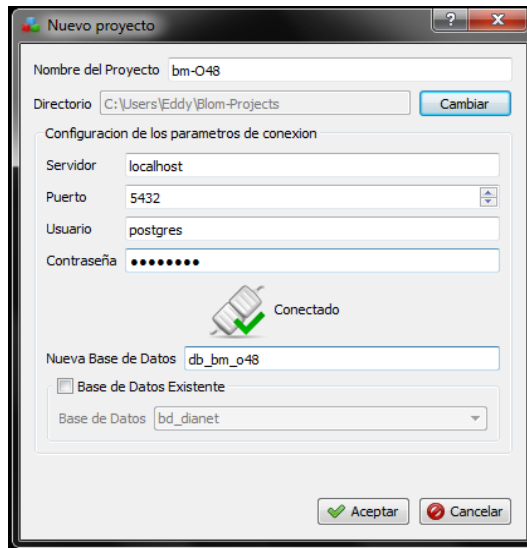


Figura 14: Crea un nuevo proyecto.

Paso 2: Crear el modelo de bloques.

En este paso se creó un modelo de bloques vacío con los datos básicos establecidos como se muestra en la **Figura 15**. El modelo tiene como nombre “*bm-O48*”.

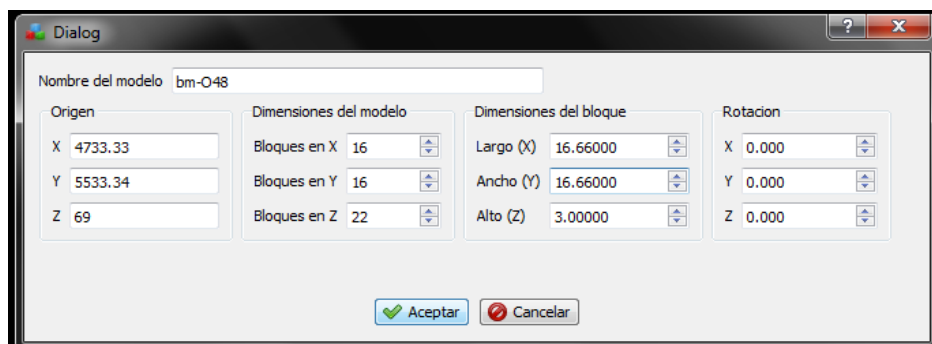


Figura 15: Crear un nuevo modelo de bloques.

Paso 3: Definir los atributos del modelo:

En este paso se le añadió al modelo los atributos (ver **Figura 16**) definidos en la **Tabla 7**.

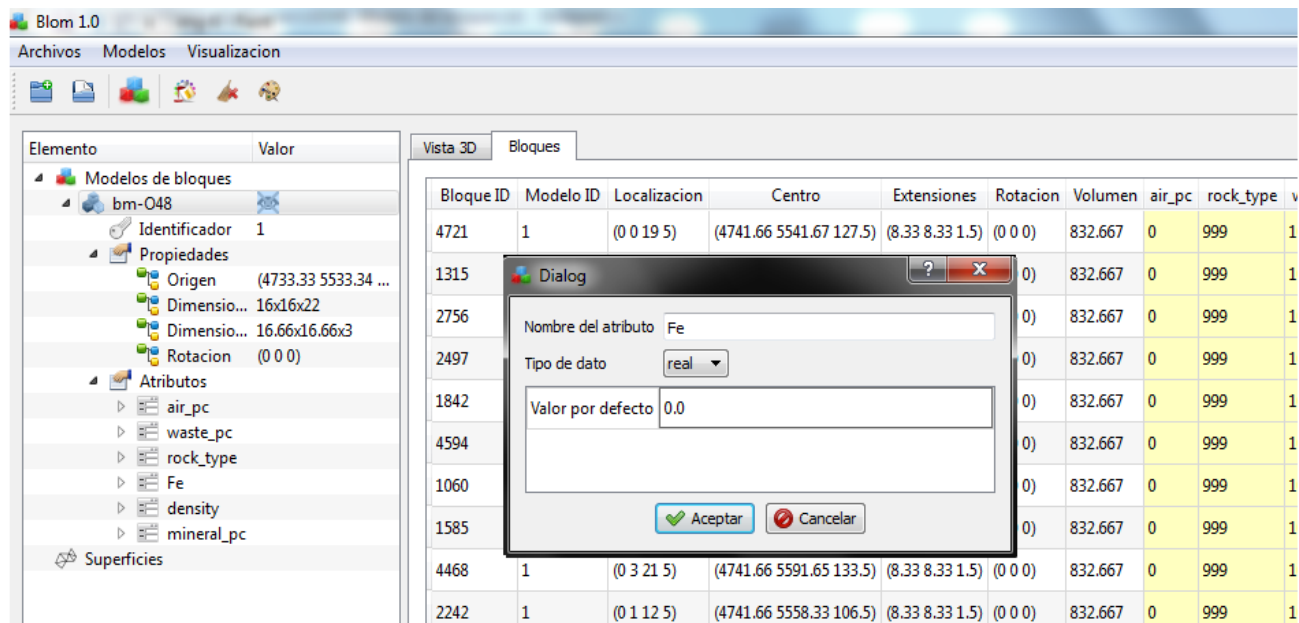


Figura 16: Atributos del modelo.

Atributos	Identificador	Tipo de datos
Hierro	Fe	Real
Densidad	Density	Real
Tipo de roca	rock_type	Entero
Porciento de Mineral	mineral_pc	Real
Porciento de Aire	air_pc	Real

Por ciento de estéril	waste_pc	Real
-----------------------	----------	------

Tabla 7: Atributos añadidos al modelo

Paso 4: Importar los datos del modelo de bloques.

En este paso se importó los datos del modelo provenientes de un fichero de texto. Estos datos fueron generados en el sistema Gemcom-Gems, donde se calculó la cantidad de hierro que tiene cada bloque del modelo y además se obtuvieron otros datos como el tipo de roca y las densidades de estas. El fichero que se importó tiene como datos las coordenadas del centro de cada bloque y los valores de los atributos del bloque (ver **Figura 17**).

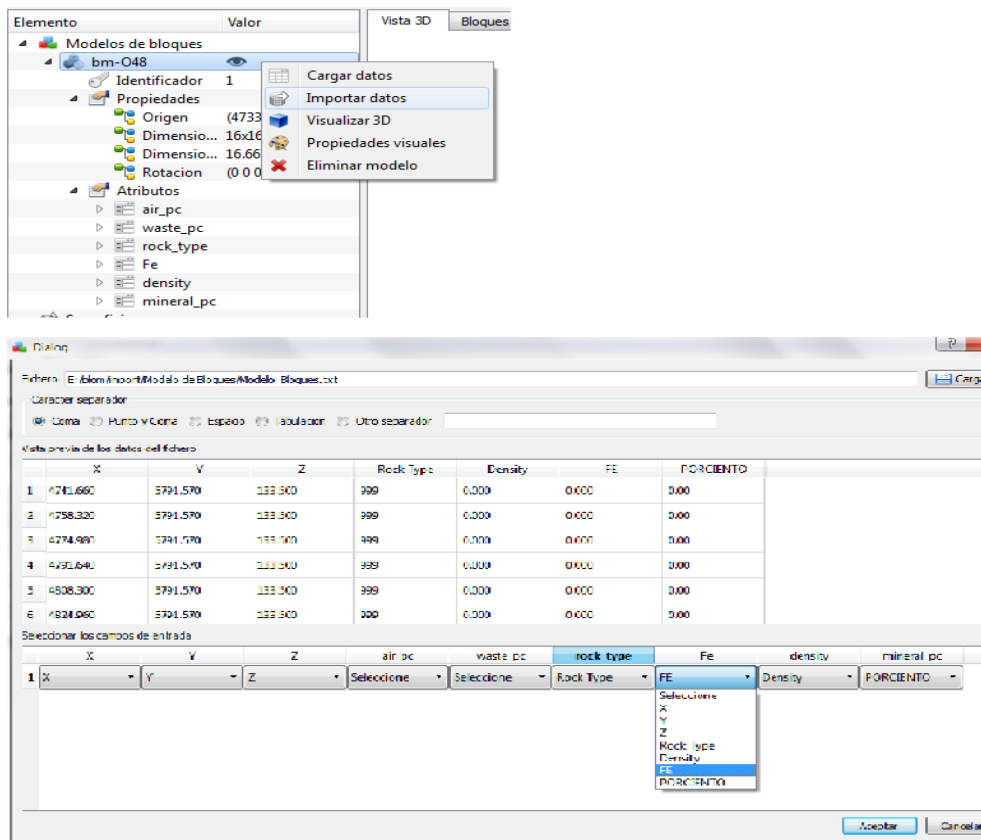


Figura 17: Importar datos del modelo.

Paso 5: Visualizar tridimensionalmente y de forma tabulada.

En este paso se visualizó tridimensionalmente bloques del modelo y se le aplicó una leyenda de colores para apreciar la variación espacial del contenido de hierro de los bloques del modelo como se muestra en la **Figura 18**. También se mostró de forma tabulada la información de los bloques del modelo.

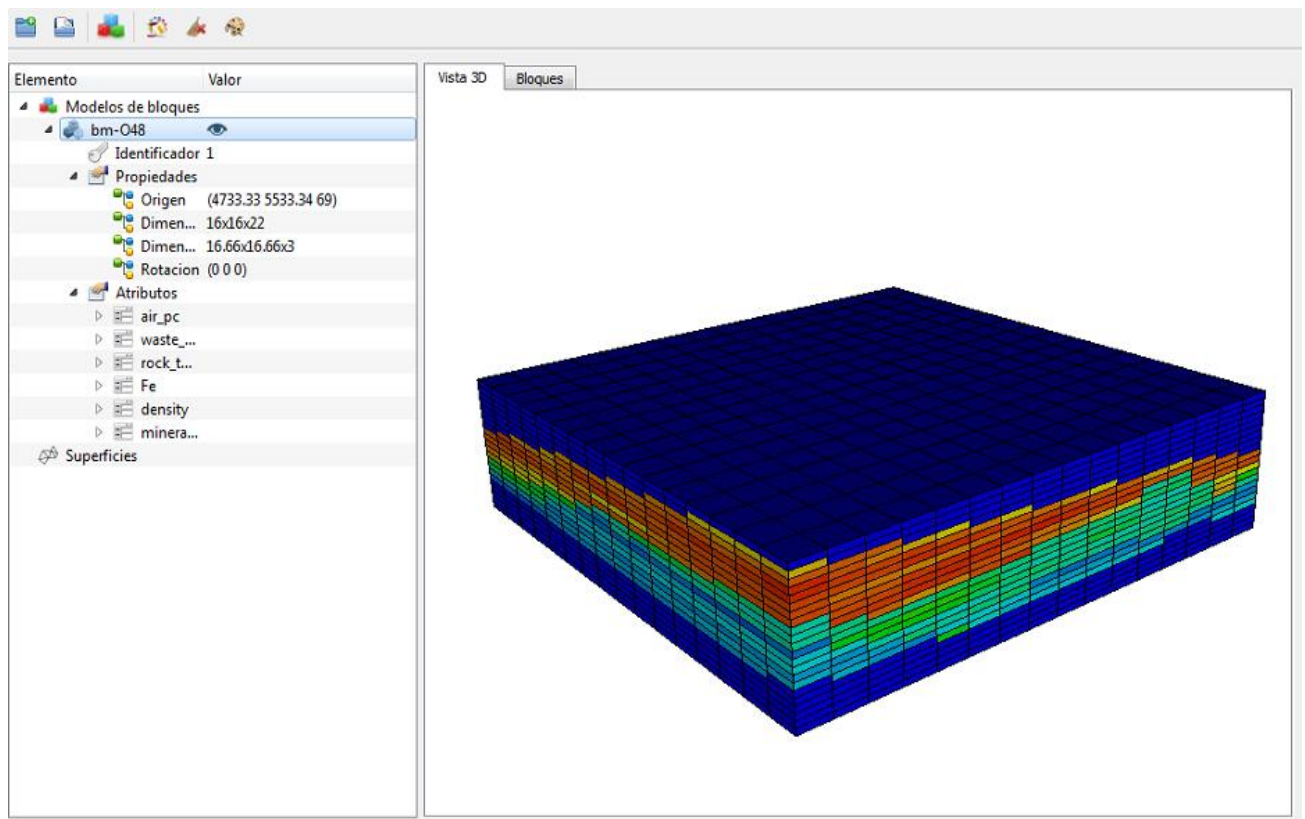


Figura 18: Visualización 3D.

4.4 Beneficios del sistema.

La herramienta informática desarrollada podrá ser desplegada en más de 20 empresas cubanas que realizan la pequeña minería y que no cuentan con el presupuesto necesario para comprar sistemas informáticos foráneos especializados en esta rama y que tienen un alto costo económico en el mercado internacional. Esto le permitirá al gobierno cubano el ahorro de miles de dólares y elevará la calidad de la gestión de los datos mineros en las empresas cubanas. Esta herramienta también servirá de base para la futura implementación del módulo Modelo de Bloques, en la aplicación informática Sistema de Análisis y Modelado de Yacimientos Minerales (Syam), desarrollado por el centro GEySED de la Universidad de las Ciencias Informáticas.

4.5 Conclusiones Parciales.

En la etapa de implementación de la solución propuesta se utilizaron las tecnologías definidas en el Capítulo 2, la arquitectura de software y los patrones de diseño definidos en el Capítulo 3, lo cual permitió que la aplicación se implementara utilizando buenas prácticas de la programación y minimizando los errores de codificación. Al concluir la etapa de implementación se obtuvo la propuesta de solución con el 100% de los requerimientos funcionales implementados y se le realizaron pruebas de caja negra logrando corregir las deficiencias funcionales detectadas en el sistema. El sistema fue probado con datos reales de la sección O48 de un yacimiento de níquel cubano demostrando así que el sistema puede ser utilizado por cualquier empresa cubana que realice la labor minera.

CONCLUSIONES GENERALES

A continuación se enuncian las conclusiones generales arribadas de la presente investigación:

- El estudio del estado del arte de los procesos de gestión y modelado de los yacimientos minerales así como el análisis de las principales soluciones existentes que realizan estos procesos permitió, sentar las bases teóricas para la futura implementación de la solución propuesta, además permitió identificar algunas de las funcionalidades básicas de la solución propuesta.
- Las herramientas y tecnologías que se escogieron para el desarrollo de la solución propuesta son de código abierto, multiplataforma y compatibles con las tecnologías utilizadas en el proyecto Sistema Minero Cubano (SMC), esta selección permitió que la propuesta de solución fuera fácil de integrar con los productos desarrollados por el proyecto SMC y que cumpliera con las políticas de migración al software libre establecidas en el país.
- La definición del diseño de la base de datos del sistema teniendo en cuenta las características del SGDB PostgreSQL y de su extensión Postgis, logró la persistencia de los datos de los modelos de bloques y permitió una mayor eficiencia en el manejo de los datos espaciales.
- La utilización de las tecnologías definidas en el Capítulo 2, la arquitectura de software y los patrones de diseño definidos en el Capítulo 3, permitió que la aplicación se implementara utilizando buenas prácticas de la programación y minimizando los errores de codificación.
- Las pruebas de caja negra realizada sobre la solución propuesta logró corregir las deficiencias funcionales detectadas en el sistema, además el sistema fue probado con datos reales de la sección O48 de un yacimiento de níquel cubano demostrando así que el sistema puede ser utilizado por cualquier empresa minera cubana.

RECOMENDACIONES

La solución propuesta cumple con el objetivo general trazado en la investigación, pero a pesar de esto la autora de la presente investigación recomienda que en futuras versiones la aplicación propuesta pueda ser complementada con otras funcionalidades como:

- Permitir la estimación de recursos minerales y la planificación de la actividad minera.
- Paralelizar las operaciones utilizando técnicas de programación paralela con los procesadores de las computadoras o con unidades de procesamiento gráfico (sus siglas en inglés GPU¹¹).

¹¹ Graphics Processing Unit: en idioma español, unidad de procesamiento gráfico.

REFERENCIAS BIBLIOGRÁFICAS

- Alarcon, Q. R. (1994). Gaceta Oficial de la República de Cuba. Ley de Minas (Ley No. 76). p 21.
- Anderson, D. J. (2004). *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*: Prentice Hall Professional.
- Avila, K., & Sobierajski, L. (2012). *The VTK User's Guide*: Kitware, Incorporated.
- Boggs, W., & Boggs, M. (2002). *Mastering UML with Rational Rose 2002*: Sybex.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The unified modeling language user guide*: Addison-Wesley.
- Burd, B. (2004). *Eclipse For Dummies*: John Wiley & Sons.
- Chen. (2004). *Developing Application Frameworks in .NET*: Apress.
- Chen, R. L., F. Dong y L. Qian. (1996). 3D Data Structures and applications in geological subsurface modeling.
- Davis, S. R. (2009). *C++ For Dummies*: John Wiley & Sons.
- Dobson, R. (2005). *Beginning SQL Server 2005 Express Database Applications with Visual Basic Express and Visual Web Developer Express: From Novice to Professional*: Apress.
- Douglas, K., & Douglas, S. (2003). *PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgreSQL Databases*: Sams Publishing.
- Duke, P. H. (2000). Geological Interpretation for Resource Estimation
- Heggoy, O. K. (2006). *Videreutvikling av open source biblioteket Coin3D*: O.K. Heggøy.
- Iznaga, J. D. A. (2001). Modelos de yacimientos minerales: tipologías y aplicaciones. [Revista]. p 4.
- Kroll, P., & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the Rup*: Addison-Wesley Professional.
- López, E. T., Ramon, A. O., Sarroca, E. M., & Seone, C. G. (2004). *Diseño de sistemas software en UML*: Univ. Politèc. de Catalunya.

- Moore, A. (2010). *Visual Studio 2010 All-in-One For Dummies*: John Wiley & Sons.
- Ning, Z. (2011). *Based on the profile of complex three dimensional stratum modeling*.
- Pipes, J., & Kruckenberg, M. (2005). *Pro MySQL*: Apress.
- Pressman, R. S. (2010). *INGENIERIA DE SOFTWARE* (2010-03-04 ed.): McGraw-Hill Interamericana de España S.L.
- Priolo, S. *Métodos Ágiles: USERSHOP*.
- Puri, C. P. (2009). *Agile Management: Feature Driven Development*: Global India Publications.
- Randolph, N., Gardner, D., Anderson, C., & Minutillo, M. (2010). *Professional Visual Studio 2010*: John Wiley & Sons.
- Rubio, E. (2008). Diseño de Minas Subterráneas. Creación de Modelos de Bloques en Datamine - Clase Auxiliar.
- Schroeder, W., Martin, K., Martin, K. W., & Lorensen, B. (1998). *The visualization toolkit*: Prentice Hall PTR.
- Sommerville, I. (2005). *Ingeniería del software*: Pearson Educación.
- Stroustrup, B. (2000). *The C++ Programming Language*: Addison-Wesley.
- Surhone, L., Tennoe, M., & Henssonow, S. F. (2010). *Qt Creator*: VDM Publishing.
- Surhone, L., Timpledon, M., & Marseken, S. (2010). *Qt (Framework)*: VDM Publishing.
- Theilin, J. (2007). *Foundations of Qt Development*: Apress.
- Troelsen, A. (2010). *Pro C# 2010 and the .NET 4 Platform*: Apress.
- Zieliński, K., & Szmuc, T. (2006). *Software Engineering: Evolution And Emerging Technologies*: IOS Press.

BIBLIOGRAFÍA

- ALARCÓN, Q. R. Gaceta Oficial de la República de Cuba. Ley de Minas (Ley No. 76), 1994: p 21.
- ANDERSON, D. J. *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall Professional, 2004. 356 p. 9780131424609
- AVILA, K. and L. SOBIERAJSKI. *The VTK User's Guide*. Kitware, Incorporated, 2012. 522 p. 9781930934238
- BOGGS, W. and M. BOGGS. *Mastering UML with Rational Rose 2002*. Sybex, 2002. 678 p. 9780782140170
- BOOCH, G.; J. RUMBAUGH, et al. *The unified modeling language user guide*. Addison-Wesley, 2005. 504 p. 9780321267979
- CABELLO, M. V. N. *Introduccion a Las Bases de Datos Relacionales*. Editorial Visión Libros, 2010. 108 p. 9788499836171
- CASTELLS, X. E. *Los residuos mineros: Reciclaje de residuos industriales*. Ediciones Díaz de Santos, 2012. 64 p. 9788499693729
- CHEN, R. L., F. DONG Y L. QIAN 3D Data Structures and applications in geological subsurface modeling, 1996.
- DAVIS, S. R. *C++ For Dummies*. John Wiley & Sons, 2009. 333 p. 9780470317266
- DINUBILE, N. A. and W. PATRICK. *FrameWork*. Rodale, 2005. 268 p. 9781609616434
- DOBSON, R. *Beginning SQL Server 2005 Express Database Applications with Visual Basic Express and Visual Web Developer Express: From Novice to Professional*. Apress, 2005. 625 p. 9781430201083
- DOUGLAS, K. and S. DOUGLAS. *PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgreSQL Databases*. Sams Publishing, 2003. 818 p. 9780735712577
- DUKE, P. H. *Geological Interpretation for Resource Estimation* 2000.
- HOULDING, S. *3D geoscience modeling: computer techniques for geological characterization*. Springer-Verlag, 1994. 332 p. 9780387580159

- IZNAGA, J. D. A. Modelos de yacimientos minerales: tipologías y aplicaciones, 2001: p 4.
- KROLL, P. and P. KRUCHTEN. *The Rational Unified Process Made Easy: A Practitioner's Guide to the Rup*. Addison-Wesley Professional, 2003. 462 p. 9780321166098
- LARMAN, C. *UML y patrones: introducción al análisis y diseño orientado a objetos*. Félix Varela, 2004. book p.
- LÓPEZ, E. T.; A. O. RAMON, et al. *Diseño de sistemas software en UML*. Univ. Politèc. de Catalunya, 2004. 198 p. 9788498800753
- LOUDEN, K. C. and K. A. LAMBERT. *Programming Languages: Principles and Practices*. Cengage Learning, 2011. 666 p. 1111529418
- MOORE, A. *Visual Studio 2010 All-in-One For Dummies*. John Wiley & Sons, 2010. 917 p. 9780470885574
- NING, Z. *Based on the profile of complex three dimensional stratum modeling*, 2011. 2847-2850 p.
- PRESSMAN, R. S. *INGENIERIA DE SOFTWARE*. 2010-03-04. McGraw-Hill Interamericana de España S.L., 2010. 959 p. 9786071503145
- PRIOLO, S. *Métodos Agiles*. USERSHOP, 2009. 27 p. 9789871347971
- RANDOLPH, N.; D. GARDNER, et al. *Professional Visual Studio 2010*. John Wiley & Sons, 2010. 1227 p. 9780470873120
- RUBIO, E. *Diseño de Minas Subterráneas. Creación de Modelos de Bloques en Datamine - Clase Auxiliar.*, 2008.
- SCHROEDER, W.; K. MARTIN, et al. *The visualization toolkit*. Prentice Hall PTR, 1998. 688 p. 9780139546945
- SMITH, M. J. and G. SAIVENDY. *Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents and Virtual Reality. of the Proceedings of HCI International 2001*. Routledge, 2001. 1608 p. 9780805836073
- SOMMERVILLE, I. *Ingeniería del software*. Pearson Educación, 2005. 714 p. 9788478290741
- STROUSTRUP, B. *The C++ Programming Language*. Addison-Wesley, 2000. 1044 p. 9780201700732
- SURHONE, L.; M. TENNOE, et al. *Qt Creator*. VDM Publishing, 2010a. p. 9786133342286

- SURHONE, L.; M. TIMPLEDON, *et al.* *Qt (Framework)*. VDM Publishing, 2010b. 100 p. 9786130999865
- THELIN, J. *Foundations of Qt Development*. Apress, 2007. 552 p. 9781430202516
- TROELSEN, A. *Pro C# 2010 and the .NET 4 Platform*. Apress, 2010. 1753 p. 9781430225492
- ZIELIŃSKI, K. and T. SZMUC. *Software Engineering: Evolution And Emerging Technologies*. IOS Press, 2006. 448 p. 9781586035594

GLOSARIO

Caso de uso (CU): Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Explotación: Conjunto de operaciones, obras, trabajos y labores mineras destinadas a la preparación y desarrollo del yacimiento y a la extracción y transportación de los minerales.

Mineral: Sustancia inorgánica que se halla en el suelo o en el subsuelo, y principalmente aquella cuya explotación ofrece interés económico.

Minería: Arte de laborar el conjunto de las minas y explotaciones mineras.

Modelación geológica: Está compuesto por celdas rectangulares o bloques donde cada una de las cuales cuenta con sus correspondientes atributos geotécnicos.

Modelo de bloques: Consiste en dividir el yacimiento en celdas tridimensionales en forma de paralelepípedos.

Requerimiento: Petición de una acción que se considera necesaria.

Software: Conjunto de instrucciones y datos codificados para ser leídas e interpretadas por una computadora. Estas instrucciones y datos fueron concebidos para el procesamiento electrónico de datos.

Yacimiento: Cualquier acumulación natural de sustancias minerales en el suelo o en el subsuelo, que pueda ser utilizado y explotado como fuente de materia prima y como fuente de energía, y las concentraciones otra sustancia mineral cuya explotación tenga importancia económica.

