



INSTITUTO SUPERIOR MINERO
METALÚRGICO DE MOA
DR. ANTONIO NÚÑEZ JIMÉNEZ

Departamento: Ingeniería Informática

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título:

Aplicación informática para la representación gráfica de mapas de isocontenido y perfil litológico en la estimación de recursos minerales.

Autor: Yosbani Bientz Aguilera

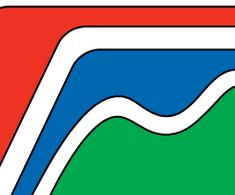
Tutor (es): MSc. Roilber Lambert

Ing. Marcos A. Martínez Rodríguez

Ing. Ivan Barea Pérez

Moa, Cuba

Julio, 2015





Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática del Instituto Superior Minero Metalúrgico de Moa para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo (firmamos) la presente a los _____ días del mes de _____ del _____.

Yosbani Bientz Aguilera

Autor

MsC. Roilber Lambert

Tutor

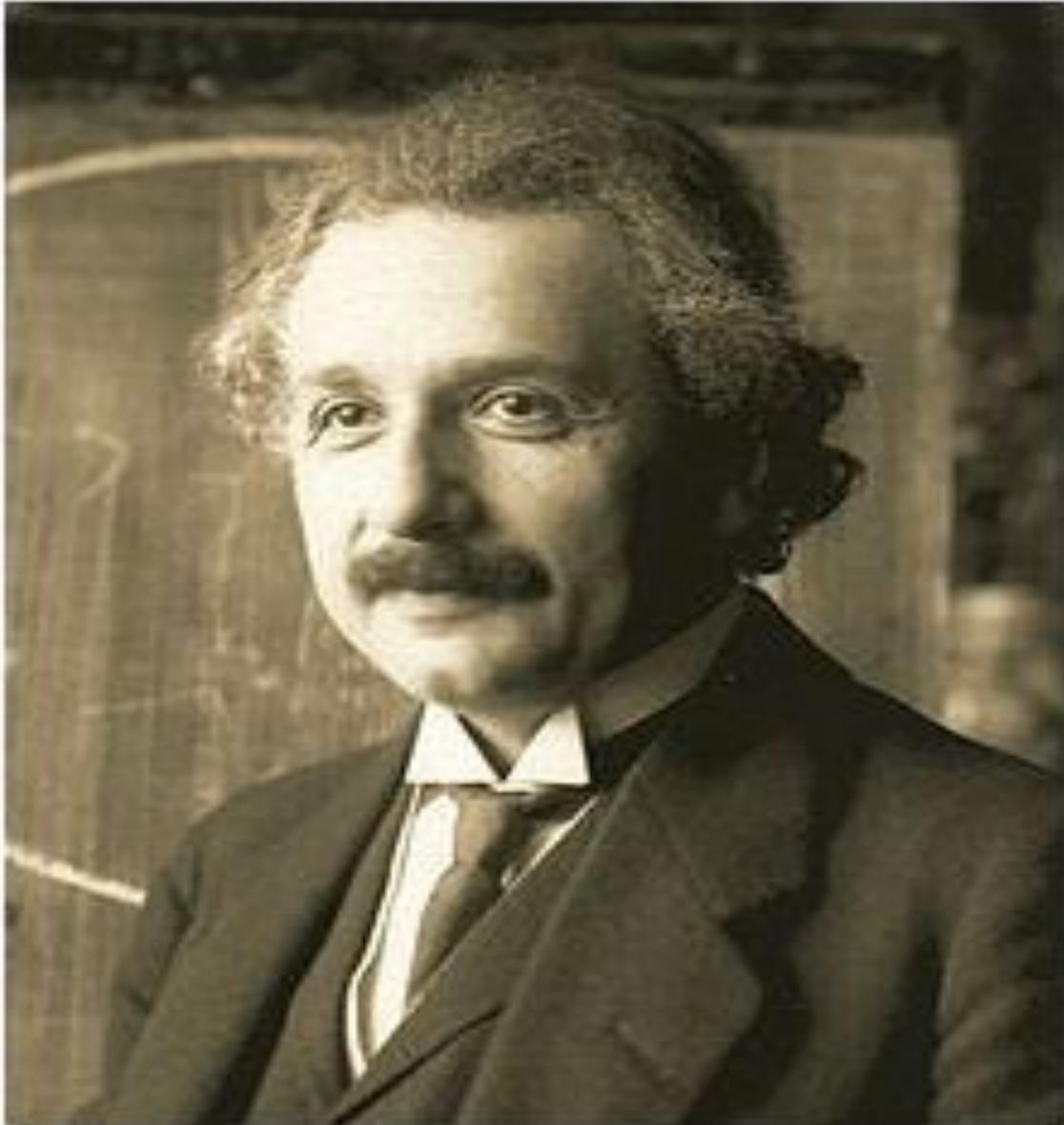
Ing. Marcos A. Martínez R

Tutor

Ing. Ivan Baréa Pérez

Tutor

PENSAMIENTO



La alegría de mirar y comprender es el don más bello de la naturaleza.

Albert Einstein.



OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: “Aplicación informática para la representación gráfica de mapas de isocontenido y perfil litológico en la estimación de recursos minerales”

Autor: Yosbani Bientz Aguilera

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <nota 2-Desaprobado, 3-Aprobado, 4-Bien, 5-Excelente>. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

(Si procede)

Nombre completo del primer tutor

Nombre completo del segundo tutor

<Grado científico, Categoría docente y/o investigativa>

<Grado científico, Categoría docente y/o investigativa>

Fecha: _____



Dedicatoria

*Este trabajo va dedicado en especial a mis padres: **Juvenal Bientz Velázquez** y **Maribel Aguilera Claro**.*

A mis familiares y amigos que me han apoyado de manera incondicional toda la vida.

Yosbani Bientz



Agradecimientos

Doy gracias:

*A Dios por darme la vida, a mis padres: **Juvenal Bientz Velázquez** y **Maribel Aguilera Claro** por formar a la persona que hoy soy.*

*A mis pastores **Orlando Hernández González** y **Yamila Regalado Marrón** que se han preocupado mucho por mí desde que me conocieron en el 2012.*

*A todos mis compañeros del aula, en especial a **José Luis**, **Yuris Ángel**, **Carlos Manuel (Perigó)**, **Orlando Prieto**, **Rolbis**, **Roger René**, **Keyler (La Malanguita)**.*

*A mis amigos incondicionales **Luis Matos (Luisi)**, **Olisday (Oli)**, **Danielkis (Nico)** y a los otros amigos que aunque no estén sus nombres escritos saben que le estoy eternamente agradecido.*

A mis demás familiares y amigos por tener esperanza en mí y ayudarme en todo lo que han podido.

*A mis tutores **Roilber Lambert**, **Marcos Martínez** y **Iván Barea**, todos los profesores que me impartieron clases, **muchas gracias**.*

Yosbani Bientz.



RESUMEN

Las Tecnologías de la Informática han jugado un papel importante en la sociedad actual, comprometiéndose en todos los aspectos del desarrollo humano. Para la gestión de los datos recolectados por los geólogos de campo, no se emplea una aplicación capaz de almacenar, modificar y representar gráficamente el avance de la perforación con el fin de detectar cualquier problema que pueda surgir durante el proceso. La investigación que se presenta en este trabajo tiene como objetivo el desarrollo de una aplicación informática para la representación gráfica de mapas que muestren la variación de los químicos y crear perfiles de litología para apoyar la estimación de recursos minerales.



ABSTRACT

The Technologies of Information Technology have played an important role in the present-day society, committing oneself in all the aspects of the human development. For the step of the data recollected by the farm geologists, a capable application to store, to modify and to represent the advance of the perforation with the aim of detecting any problem that may happen in the process graphically is not used. The investigation that presents itself in this work aims at the development of an information-technology application for the graphic performance of maps that they show the variation of the chemicals and creating profiles of lothology to back up the esteem of mineral resources.



ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
Introducción	5
1.1 Estado del Arte	5
1.2 Conceptos fundamentales utilizados en esta investigación	6
1.3 Flujo Actual de los procesos	6
1.4 Análisis Crítico de la ejecución de los procesos.	7
1.5 Antecedentes	7
Características de estos sistemas	8
1.6 Modelo Geométrico para la representación de los Pozos	9
1.6.1 Diagramas de Voronoi	9
1.7 Tendencias y tecnologías actuales	10
1.7.1 Microsoft Studio Express 2013	10
1.7.2 OpenGL	11
1.7.3 Lenguajes de Programación.....	12
1.7.4 Gestores de Base de Datos	16
1.7.5 Patrones Arquitectónicos	19
1.7.5.1 Arquitectura en capas.....	20
1.7.5.2 Modelo Cliente – Servidor de 2 capas	21
1.7.5.3 Modelo – Vista – Controlador	22
1.7.5.4 Arquitectura Propuesta para el desarrollo de la aplicación	24
1.7.6 Rational Rose Enterprise.	24
1.7.7 Embarcadero ER/Studio	24
1.7.8 Fundamentación de la selección de las Herramientas	25
1.7.8.1 ¿Por qué seleccionar C#?	25
1.7.8.2 ¿Por qué seleccionar SQLite?	25
1.9 Metodología de desarrollo de software	26
1.9.1 Metodologías Agiles.....	26
Conclusiones	30
CAPÍTULO 2: REQUISITOS	31
2.1 Introducción	31
2.2 Breve descripción de los procesos a informatizar	31
2.3 Actores del Sistema.	31



2.4 Funcionalidades del Sistema	31
2.5 Diagrama de caso de usos del sistema.....	33
2.6 Características del sistema.....	34
2.7 Historias de Usuario.	35
2.9 Planificación de entregas	37
2.9.1 Planificación de iteraciones	37
2.9.2 Estimación del esfuerzo por historias de usuario	38
2.9.3 Plan de duración de las iteraciones	39
2.9 Diseño.....	40
2.9.1 Tarjetas CRC.....	40
Conclusiones.....	43
Capítulo 3 DESCRIPCIÓN Y PRUEBA DE LA SOLUCIÓN PROPUESTA	44
3.1 Introducción.....	44
3.2 Diagrama de clases del diseño.	44
3.3 Modelo de Datos	45
3.4 Interfaz de usuario.	47
3.5 Ayuda	49
3.6 Tratamientos de errores.	49
3.7 Diagramas de Secuencias.....	49
3.8 Diagrama de Componente	51
3.9 Tareas de ingeniería	52
3.9.1 Pruebas.....	52
3.9.1.1 Pruebas de Aceptación	53
Conclusiones.....	54
Capítulo 4- Estudio de Factibilidad	55
4.1. Introducción.	55
4.2. Efectos Económicos.....	55
4.3 Beneficios y Costos intangibles en el proyecto:	57
4.4 Fichas de Costo:	57
Conclusiones del capítulo.	61
CONCLUSIONES GENERALES.....	62
RECOMENDACIONES.....	63
Referencias	64



Bibliografía 66

Anexo 1 Historias de Usuario 68



ÍNDICE DE FIGURAS

Fig. 1.1: Arquitectura en Capa	20
Fig. 1.2 Elementos de la arquitectura en dos capas	22
Fig. 1.3 Patrón Arquitectura Modelo Vista Controlador	23
Fig.2.1 Diagrama caso de uso del sistema	33
Fig. 3.1 Diagrama de clase de Diseño	45
Fig. 3.2 Modelo de Datos	46
Fig. 3.3 Interfaz para el caso de uso Gestionar Pozo	47
Fig. 3.4 Interfaz del caso de uso Gestionar Muestra	47
Fig. 3.5 Interfaz del caso de uso Gestionar Quimismo	48
Fig. 3.6 Interfaz Principal	48
Fig. 3.7 Diagrama de secuencia Insertar Pozos	50
Fig. 3.8 Diagrama de secuencia Insertar Muestra	50
Fig. 3.9 Diagrama de Componente	51
Fig. 4.1 Gráfica de comparación de la solución sin proyecto y con proyecto	60
Fig. 5.1 Diagrama de Secuencia Modificar Pozo	85
Fig. 5.2 Diagrama de Secuencia Eliminar Pozo	85
Fig. 5.3 Diagrama de Secuencia Modificar Muestra	86
Fig. 5.4 Diagrama de Secuencia Eliminar Muestra	86
Fig. 5.5 Diagrama de Secuencia Insertar Geólogo	87
Fig. 5.6 Diagrama de Secuencia Modificar Geólogo	87
Fig. 5.7 Diagrama de Secuencia Eliminar Geólogo	88
Fig. 5.8 Diagrama de Secuencia Insertar Quimismo	88
Fig. 5.9 Diagrama de Secuencia Modificar Quimismo	89
Fig. 6.1 Diagrama de Secuencia Eliminar Quimismo	89
Fig. 6.2 Diagrama de Secuencia Crear Gráfica 2D	90
Fig. 6.3 Diagrama de Secuencia Crear Gráfica 3D	90
Fig. 6.4 Diagrama de Secuencia Crear Perfil Litológico	91
Fig. 6.5 Diagrama de Secuencia Exportar	91
Fig. 6.6 Diagrama de Secuencia Importar Pozo	92
Fig. 6.7 Diagrama de Secuencia Importar Muestra	92
Fig. 6.8 Diagrama de Secuencia Importar Quimismo	93



ÍNDICE DE TABLA

Tabla 2.1 Descripción de los actores del negocio	31
Tabla 2.2 Lista de Funcionalidades del Sistema	32
Tabla 2.3 Lista de Características del Sistema	34
Tabla 2.4 HU Gestionar Pozo	36
Tabla 2.5 HU Gestionar Muestra	37
Tabla 2.6 Estimación de esfuerzos por historias de usuario	39
Tabla 2.7 Plan de Iteraciones (en Semanas)	39
Tabla 2.8 Tarjeta CRC del caso de uso Gestionar Pozo	41
Tabla 3.1 Tarea de programación Insertar Pozo	52
Tabla 3.2 Prueba de Aceptación para la HU Gestionar Pozo	53
Tabla 4.1 Costo en Moneda Librementemente Convertible	57
Tabla 4.2 Costo en Moneda Nacional	58
Tabla 5.1 HU Gestionar Quimismo	68
Tabla 5.2 HU Gestionar Geólogo	68
Tabla 5.3 HU Crear Gráfico 2D	68
Tabla 5.4 HU Crear Gráfico 3D	69
Tabla 5.5 HU Crear Perfil Litológicos	69
Tabla 5.6 HU Importar	70
Tabla 5.7 HU Exportar	70
Tabla 5.8 Tarjeta CRC del caso de uso Gestionar Geólogo	71
Tabla 5.9 Tarjeta CRC del caso de uso Gestionar Geólogo	72
Tabla 6.1 Tarjeta CRC del caso de uso Gestionar Quimismo	72
Tabla 6.2 Tarjeta CRC del caso de uso Crear Gráfica 2D	73
Tabla 6.3 Tarjeta CRC del caso de uso Crear Gráfica 3D	73
Tabla 6.4 Tarjeta CRC del caso de uso Crear Perfil Litológico	74
Tabla 6.5 Tarjeta CRC del caso de uso Importar	74
Tabla 6.6 Tarjeta CRC del caso de uso Exportar	74
Tabla 6.7 Prueba del HU Gestionar Muestra	76
Tabla 6.8 Prueba del HU Gestionar Quimismo	76
Tabla 6.9 Prueba del HU Gestionar Geólogo	76
Tabla 7.1 Prueba del HU Crear Gráfica 2D	77
Tabla 7.2 Prueba del HU Crear Gráfica 3D	77
Tabla 7.3 Prueba del HU Crear Perfil Litológico	78
Tabla 7.4 Prueba del HU Importar	78



Tabla 7.5 Prueba del HU Exportar	78
Tabla 7.6 Tarea de Ingeniería Insertar Geologo	80
Tabla 7.7 Tarea de Ingeniería Modificar Geologo	80
Tabla 7.8 Tarea de Ingeniería Eliminar Geologo	80
Tabla 7.9 Tarea de Ingeniería Modificar Pozo	80
Tabla 8.1 Tarea de Ingeniería Eliminar Pozo	81
Tabla 8.2 Tarea de Ingeniería Insertar Muestra	81
Tabla 8.3 Tarea de Ingeniería Modificar Muestra	82
Tabla 8.4 Tarea de Ingeniería Eliminar Muestra	82
Tabla 8.5 Tarea de Ingeniería Insertar Quimismo	82
Tabla 8.6 Tarea de Ingeniería Modificar Quimismo	82
Tabla 8.7 Tarea de Ingeniería Eliminar Quimismo	83
Tabla 8.8 Tarea de Ingeniería Crear Grafica 2D	83
Tabla 8.9 Tarea de Ingeniería Crear Grafica 3D	83
Tabla 9.1 Tarea de Ingeniería Crear Perfil Litologico	83
Tabla 9.2 Tarea de Ingeniería Importar	84
Tabla 9.3 Tarea de Ingeniería Exportar	84



INTRODUCCIÓN

El Instituto Superior Minero-Metalúrgico de Moa “Dr. Antonio Núñez Jiménez” (ISMMM) es la principal institución académica del país en la rama Geólogo-Minera y Metalúrgica, desde su creación se ha convertido en un importante eslabón en el desarrollo de la Industria Cubana del Níquel. (1)

La relevancia que confiere el estado al progreso de la Informática y la industria del software se explicitó en la Resolución Económica del V Congreso del Partido Comunista de Cuba (PCC) en octubre de 1997:

- En particular las industrias (...), incluida la del software, deberán trabajar con calidad, por incrementar la sustitución de importaciones, y desarrollar la exportación de líneas de elevada eficiencia.
- El país debe encaminarse resueltamente a la modernización informática mediante un programa integral que involucre a las organizaciones que deben proveer los recursos materiales, financieros e intelectuales y a las entidades económicas, políticas y sociales que deben traducirlos en más y mejores productos y servicios.

En consecuencia el perfil de la carrera de Ingeniería Informática se ha encaminado al progreso y perfeccionamiento de la economía del país, con el objetivo de hacer de las instituciones, lugares donde premie la eficiencia y la eficacia, a través de la vinculación de la Tecnologías de la Informática y las Comunicaciones (TIC). El Ingeniero Informático debe desempeñar un rol en el que facilite la dirección de las empresas mediante la automatización de los sistemas de información, la incorporación de diferentes programas informáticos que sirvan de guía, ayuda en el trabajo y en la toma de decisiones para la gestión de procesos tecnológicos. Además de proveer una serie de aplicaciones que contribuya en mejores servicios y productos factibles al mejoramiento continuo del país.

Los profesionales deben ser capaces de enfrentar los retos que impone en la sociedad donde las tecnologías ocupan un papel relevante en todas las esferas, por lo que el



desarrollo de habilidades, el uso de técnicas para el procesamiento y uso de la información junto a la creación de mejores herramientas que favorezcan el trabajo permiten transmitir experiencias y conocimientos.

Actualmente los departamentos del Centro Internacional de la Habana (CIH), Geología y Minas ubicados en el ISMMM trabajan en los proyectos de estudio de yacimientos para la estimación de recursos minerales para su posterior explotación por la fábrica niquelífera Pedro Soto Alba. Para la gestión de los datos registrados por los geólogos de campo no se emplea una aplicación capaz de almacenar y modificar estos, ni representar gráficamente el avance de la perforación, con el fin de detectar cualquier problema que pueda surgir durante el proceso. Los cálculos y procedimientos se desarrollan con el empleo de Bases de Datos de Microsoft Access o mediante hojas de cálculo de Microsoft Excel. Por otra parte, no existe una correspondencia entre los valores litológicos (horizontes) observados por los expertos en la toma de muestras y los resultados que arrojan las pruebas de quimismo.

Por lo que se define como **problema científico**: ¿Cómo favorecer el estudio de yacimientos minerales para la estimación de recursos? Definiendo el **objeto de estudio**: informatización del estudio de yacimientos minerales, enmarcada en el **campo de acción**: representación gráfica de mapas de isocontenido y perfil litológico. Para dar solución al problema se propone como **objetivo general**: Crear una aplicación informática que apoye el estudio de yacimiento para la estimación de recursos minerales mediante la representación gráfica de mapas de isocontenido y perfil litológico.

Se definen como **objetivos específicos**:

- Confeccionar el estado del arte de la investigación.
- Diseñar e implementar la aplicación informática para la representación gráfica de mapas de isocontenido y perfil litológico en la estimación de recursos minerales
- Realizar un estudio de factibilidad.

Se definen como **tareas de la investigación**:



- Estudio de las aplicaciones similares en el mundo y en Cuba para identificar procesos afines con el sistema a desarrollar.
- Selección de las tecnologías y metodologías para el desarrollo de la aplicación.
- Definición de las funcionalidades que va a contener la aplicación.
- Definición de los requerimientos de software para cumplir con las exigencias del cliente.
- Diseño de una arquitectura que satisfaga los objetivos planteados.
- Implementación del sistema.
- Elaboración y aplicación de pruebas al sistema para comprobar que cumpla con los requisitos funcionales.
- Evaluación de los costos y beneficios que demuestre la rentabilidad y factibilidad del sistema.

Para dar cumplimiento al objetivo general se plantea la siguiente **idea a defender**: Si se realiza la aplicación informática para la representación gráfica de mapas de isocontenido y perfil litológico, se favorecerá los estudios de yacimientos minerales.

Entre los **métodos de investigación** utilizados se encuentran los *teóricos* y los *empíricos*. Ambos arquetipos se enumeran a continuación:

Método Empírico

- **Experimento**: para realizar pruebas al sistema y verificar el cumplimiento de cada uno de los requisitos funcionales acordados.

Métodos Teóricos

- **Histórico-lógico**: permitió conocer los antecedentes del tema a investigar, si existen otras aplicaciones informáticas para apoyar la aplicación, características que poseen.



- **Analítico – Sintético:** para estudiar todo el contenido en los documentos revisados, sintetizar, clasificar y evaluar la información valiosa, logrando una mejor comprensión del sistema.
- **Hipotético-deductivo:** a través de este método se realizó el correcto análisis para el desarrollo de la herramienta que sustentará la idea a defender y guiará el proceso hasta darle solución al problema planteado.

El trabajo consta del Resumen, Introducción, 4 Capítulos, Conclusiones, Recomendaciones, Referencias, Bibliografía y Anexos:

Capítulo 1-Fundamentación Teórica: Brinda una descripción de los conceptos relacionados al objeto de estudio, el objetivo general y las tareas específicas, así como un análisis de los sistemas existentes vinculados al campo de acción, también se abordan temas específicos relacionados con la metodología y las herramientas a utilizar para el desarrollo del software.

Capítulo 2-Requisitos: Se hace uso de la metodología expuesta en el capítulo anterior para el desarrollo del proyecto, abordando en detalles cada una de sus fases, Se define los actores del sistema, las funcionalidades, el diagrama de caso de uso del sistema, las características que tendrá el sistema, las historias de usuarios, los planes de iteración y las tarjetas CRC.

Capítulo 3-Descripción de la solución propuesta: Se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. Se describen además las pruebas realizadas y sus resultados.

Capítulo 4-Estudio de Factibilidad: Se hace un análisis de su factibilidad y un gráfico de comparación de ventajas de la realización del proceso de la forma manual y de la forma informatizada.

Bibliografías y anexos: Se muestra las bibliografías consultadas para la elaboración del producto final obtenido y anexos que respalden la metodología utilizada para el desarrollo del trabajo.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se presenta una descripción general del escenario donde radica el objeto de estudio, así como el campo de acción en que se enmarca la investigación. Se presenta una mejor visión sobre la herramienta a desarrollar. Este capítulo constituye la base teórica para la comprensión del trabajo que se desarrolla. Se abordarán conceptos asociados al análisis de las tecnologías y herramientas a utilizar, además de la metodología seleccionada para el desarrollo del software.

1.1 Estado del Arte

El empleo de mapas en las investigaciones científicas data de los siglos pasados. Su empleo en la navegación, en la guerra, e incluso en la medicina ha sido amplio. Por ejemplo cuando en el siglo XVIII el francés Louis Alexander Berthier, mostró el posible desplazamiento de las tropas militares a través de la superposición de mapas o en el siglo XIX en la ciudad de Londres el médico inglés Bell determinó el radio de influencia de la peste, realizando “buffer” (área de influencia) alrededor de los pozos de agua.

Ya para el siglo XX en la década del 60, producto del desarrollo de diferentes disciplinas tales como la cartografía, la fotogrametría, así como el rápido desarrollo de los sistemas de cómputo y la revolución del análisis cuantitativo espacial, se creó el primer **SIG**¹ (Sistema de Información geográfico), cuyo principal objetivo fue el inventario de terrenos (CGIS). Cuando Tomlinson, Calking y Marble desarrollaron el Sistema Canadiense (Canadian Geographical Information System) conocido por sus siglas CGIS.

¹ **SIG**: Según el National Center for Geographic Information and analysis de U.S.A., 1990 un SIG no es más que un sistema de Hardware, software y procedimientos, elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos especialmente referenciados, para resolver problemas complejos de planificación y gestión. (35)



1.2 Conceptos fundamentales utilizados en esta investigación

- Litología: es la parte de la geología que trata de las rocas, especialmente de su tamaño de grano, del tamaño de las partículas y de sus características físicas y químicas. (2)
- Perfil Litológico o columna litológica: representa gráficamente la estratigrafía² generalizada de un sector. (2)
- Perfil o corte geológico: puede definirse como una sección vertical o perfil interpretativo de la geología superficial, para cuya realización se utilizan los datos obtenidos en el terreno. Es decir, un corte geológico es la interpretación de la información geológica disponible de una zona, representada en un corte o sección. (2)
- Isocontenido: Variación de la concentración de un químico en un plano. (2)
- Muestra: Es aquella porción de mineral o roca que contiene o posee en determinada medida, los parámetros, calidades, leyes o tenores que definen a la masa del yacimiento o sus rocas de caja, que puede ser o no representativa del mismo, obtenidas por diferentes vías, y estar compuesta por partículas de diferentes diámetros y peso variable, en dependencia de los objetivos para los que se tome. (3)

1.3 Flujo Actual de los procesos

El CIH y los Departamentos de Geología y Minas ubicados en el ISMMM, trabajan en los proyectos de estudio de yacimientos para la estimación de recursos minerales, para su posterior explotación. En el proceso de perforación, lo primero que se realiza es la ubicación de las coordenadas (login) de los pozos, luego se hace un plano de todos los pozos ubicados, se comprueba el área, se hace las correcciones de los pozos y se elimina el material vegetal de las áreas, luego se perforan los pozos. De cada pozo se necesita registrar los datos del pozo, la cantidad de muestras, de las muestras su

²**estratigrafía** Parte de la geología que estudia la disposición y caracteres de los estratos o capas de la corteza terrestre.



quimismo. Actualmente estos datos se guardan en tablas estáticas (Excel) y bases de datos relacionales (Access), además de que las gráficas de las áreas y la graficación de los perfiles litológicos se deben de hacer manualmente para su consulta.

1.4 Análisis Crítico de la ejecución de los procesos.

El proceso de almacenamiento de la información y su graficación se realiza de forma manual y descentralizada lo cual dificulta el trabajo de los especialistas ya que la graficación no puede ser exacta al no contar con la correspondencia de los valores químicos con los valores litológicos (horizontes). Para los geólogos se les hace muy difícil detectar errores de coordenadas a la hora de gestionar los datos de los pozos y a la hora de hacer las áreas de perforación.

1.5 Antecedentes

Internacionales

ARC/INFO: En el Instituto de Investigaciones de Sistemas Ambientales de California (ESRI), en 1982 surge el ARC/INFO, que hoy en día constituye el paquete comercial de SIG más extendido a escala mundial. Actualmente existen miles de paquetes en todo el mundo, que abarcan un gran espectro de orientación que va desde el control de recursos de municipalidades, análisis de redes, hasta el trabajo más ambicioso que se pueda generar en el mundo de las Geociencias. (4)

Mapinfo: es una herramienta de creación de mapas. Las soluciones que proporciona la creación de mapas por Computadora están a su alcance, permite llevar a cabo análisis geográficos sencillos y complejos, acceso a datos remotos, creación de mapas temáticos que revelen patrones en los datos. Puede visualizar los datos como puntos, como regiones zonificadas temáticamente, como gráficos de tarta o de barras, etc. Puede llevar a cabo operaciones de zonificación, combinación y división de objetos, y definición de áreas de influencia. También puede realizar consultas acerca de los datos. (5)

ARGIS: es el nombre de un conjunto de productos de software en el campo de los Sistemas de Información Geográfica o SIG. Producido y comercializado por ESRI, bajo el nombre genérico ArcGIS se agrupan varias aplicaciones para la captura, edición,



análisis, tratamiento, diseño, publicación e impresión de información geográfica. Estas aplicaciones se engloban en familias temáticas como *ArcGIS Server*, para la publicación y gestión web, o *ArcGIS Móvil* para la captura y gestión de información en campo. Si bien es uno de los mejores software de Geociencias, es uno de los más caros en el mundo debido a sus aplicaciones. (6)

Quantum GIS: es un SIG de código libre para plataformas GNU/Linux, Unix, Mac OS y Microsoft Windows. Era uno de los primeros ocho proyectos de la Fundación OSGeo. Permite manejar formatos raster y vectoriales a través de las bibliotecas GDAL y OGR, así como bases de datos. (7)

Características de estos sistemas.

- 1- Son reconocidos mundialmente dentro del entorno minero.
- 2- Son software propietario y su adquisición es costosa.
- 3- Empleados para proyectos mineros de alto nivel.
- 4- Manejan grandes cantidades de datos, más de los requeridos para la representación gráfica de mapas de isocontenido y perfil litológico en la estimación de recursos minerales.

Nacionales

Cuba no ha quedado exenta de este desarrollo, es así que por el año 1987 surge en el Instituto de Geografía de la Academia de Ciencias de Cuba el “**SIG de Cuba**” con el objetivo fundamental de actualizar el Atlas Nacional de Cuba, tomándolo como fuente inicial de datos y como receptor final. A finales de la década del 80 Aronoff (1989) realizó un conjunto de procedimientos manuales o computarizados usados para almacenar y tratar datos referenciados geográficamente.

Por su parte, el Departamento de Computación y Matemática Aplicada del Instituto Cubano de Hidrografía (hoy GEOCUBA) desarrolló a partir del año 1990 el producto **TeleMap** que en su versión actual constituye una herramienta muy poderosa para el diseño de los SIGs y que ha sido ampliamente generalizado en todo el país. Su existencia le permite a los especialistas que en el territorio nacional necesitan utilizar la



tecnología SIG, poder contar con una opción real y económica, que además le permite acceder a la asesoría necesaria para su instalación y manipulación.

En nuestro Instituto (ISMMM) se han desarrollado software similares para dar respuesta a disímiles problemas de empresas ejemplo el Tierra desarrollado por el Dr. Arístides Legrá y MGCantera 1.0 de la Ing. Katusca Jiménez Roche. Las soluciones para muchos problemas frecuentemente requieren acceso a varios tipos de información que solo puede ser relacionada por geografía o información espacial. Solo la tecnología SIG permite almacenar, manipular la información geográfica, para analizar patrones, relaciones y tendencias en la información, todo ello para tomar mejores decisiones.

1.6 Modelo Geométrico para la representación de los Pozos.

Un modelo geométrico es aquel que caracteriza espacialmente al objeto estudiado, en este caso, a la porción del macizo rocoso (el yacimiento o bloque) en el cual están contenidos sus elementos esenciales (principales propiedades y estructura). El método matemático utilizado en esta investigación para la obtención del modelo geométrico fue el de triangulación de Delaunay. La triangulación de Delaunay maximiza los ángulos interiores de los triángulos de la triangulación, el método consiste en el cálculo gráfico del diagrama de Voronoi para los puntos y la posterior extracción de la triangulación a partir de este, aprovechando que la triangulación de Delaunay es el grafo dual del diagrama de Voronoi. Eso es muy práctico porque al usar la triangulación como modelo tridimensional los errores de redondeo son mínimos. Por eso, en general se usan triangulaciones de Delaunay en aplicaciones gráficas. (8)

1.6.1 Diagramas de Voronoi

La técnica clásica para la generación de diagramas de Voronoi de una nube de puntos de forma gráfica, permite generar una imagen en la que cada pixel contiene un color que representa al punto de la nube más cercano a ese pixel. Para ello se genera en tres dimensiones un cono por cada punto, y se dibuja en vista ortográfica superior, utilizando el algoritmo de Z-bufer para determinar qué cono resulta el más cercano a la cámara en cada pixel. Esta aproxima los conos utilizando triángulos, con la consiguiente pérdida de



precisión en la superficie, y los problemas de rendimiento provocados por tener que aumentar la cantidad de polígonos por cada cono para reducir los errores en el Voronoi. Permite dibujar superficies que se comportan como conos perfectos en el Z-bufer, y presentan un rendimiento excelente. Una variante a esta técnica consiste en dibujar un cuadrado en el lugar de cada cono y calcular, en cada pixel cubierto por el cuadrado, la profundidad exacta que tendría un cono perfecto. Para esto se calcula en cada pixel la distancia desde ese pixel hasta el punto de la nube al que corresponde, y se asigna ese valor como profundidad Z del pixel. A continuación, el Z-bufer se encarga de seleccionar la superficie cuyo punto central sea el más cercano a ese pixel. (9)

1.7 Tendencias y tecnologías actuales

1.7.1 Microsoft Studio Express 2013

Microsoft Visual Studio Express disponibles de forma gratuita, es un entorno de desarrollo integrado que permite diseñar, desarrollar, depurar e implantar con rapidez soluciones basadas en el .NET Framework, con un conjunto de potentes herramientas, diseñadores y editores desde cualquiera de los lenguajes de programación de Visual Studio .NET que simplifican todo el proceso de desarrollo de aplicaciones. (10)

Cuenta con una serie de nuevas características que hacen más fácil el trabajo de los desarrolladores, algunas de las más significativas son:

- Incorpora la herramienta *IntelliTrace*³, la cual permite depurar la aplicación en tiempo de ejecución, archivando errores enriquecidos y modificables para que los desarrolladores puedan reproducir siempre el error del que se informe en el estado en el que se encontró. Además cuenta con analizadores de código estático, métricas de código y creación de perfiles. (11)
- Cuenta con pruebas de UI⁴ automatizadas para la realización de interfaz de usuario en aplicaciones basadas en Web y de escritorio, así como de pruebas manuales, de

³ **IntelliTrace:** Herramienta para trazar el código.

⁴ **UI:** User Interface. Interface de Usuario.



rendimiento Web, de carga, cobertura de código y otras características completas que no se encuentran en otras ediciones de Visual Studio. (10)

- Cuenta con un entorno de desarrollo integrado que permite organizar el IDE⁵ en varios monitores, el desacoplamiento de ventanas fuera de la interfaz de trabajo, además de la característica Box Selection para aplicar cambios a múltiples líneas de código. (12)

1.7.2 OpenGL

OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD⁶, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compete con Direct3D en plataformas Microsoft Windows.

OpenGL tiene dos propósitos esenciales:

- Ocultar la complejidad de la interfaz con las diferentes tarjetas gráficas, presentando al programador una API⁷ única y uniforme.
- Ocultar las diferentes capacidades de las diversas plataformas hardware, requiriendo que todas las implementaciones soporten la funcionalidad completa de OpenGL (utilizando emulación software si fuese necesario).

El funcionamiento básico de OpenGL consiste en aceptar primitivas tales como puntos, líneas y polígonos, y convertirlas en píxeles. Este proceso es realizado por una pipeline

⁵ **IDE:** Integrated Development Environment, Entorno de Desarrollo Integrado.

⁶ **CAD:** Diseño Asistido por Computadora.

⁷ **API:** (Application Programming Interface) Interfaz de Programación de Aplicaciones



gráfica conocida como Máquina de estados de OpenGL. La mayor parte de los comandos de OpenGL emiten primitivas a la pipeline gráfica o bien configuran cómo la pipeline procesa dichas primitivas. Hasta la aparición de la versión 2.0 cada etapa de la pipeline ejecutaba una función prefijada, resultando poco configurable. A partir de la versión 2.0 algunas etapas son programables usando un lenguaje de programación llamado GLSL. (13)

OpenGL es una API basada en procedimientos de bajo nivel que requiere que el programador dicte los pasos exactos necesarios para renderizar una escena. Esto contrasta con las APIs descriptivas, donde un programador sólo debe describir la escena y puede dejar que la biblioteca controle los detalles para representarla. El diseño de bajo nivel de OpenGL requiere que los programadores conozcan en profundidad la pipeline gráfica, a cambio de darles libertad para implementar algoritmos gráficos novedosos. (13)

OpenGL ha influido en el desarrollo de las tarjetas gráficas, promocionando un nivel básico de funcionalidad que actualmente es común en el hardware comercial; algunas de esas contribuciones son:

- Primitivas básicas de puntos, líneas y polígonos rasterizados.
- Proceso en la pipeline de gráficos.
- Una pipeline de transformación e iluminación.
- Z-buffering.
- Mapeado de texturas.
- Alpha blending. (13)

1.7.3 Lenguajes de Programación

¿Qué es un lenguaje de Programación?

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse



para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. (14)

Los lenguajes de programación de alto nivel, facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas. Son herramientas que nos permiten crear software. Estos proporcionan un marco conceptual necesario para el desarrollo, análisis, optimización y comprensión de los programas y, en general, de las tareas de programación. Entre ellos tenemos Delphi, Visual Basic, Pascal, Java, C#, C++, C, etc.

C++

C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C y le agrega la capacidad de abstraer tipos como en Smalltalk. Proporciona soporte a clases, objetos, herencia, es decir, todo lo característico de un lenguaje orientado a objetos. Estas cualidades facilitan la reutilización de código, consiguiendo con ello un ahorro de trabajo y tiempo. Una de sus principales características está dada en el soporte de plantillas o programación genérica (templates). Posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel como es la posibilidad de redefinir los operadores (sobrecarga de operadores) y la identificación de tipos en tiempo de ejecución. Posee un entorno de desarrollo simple, flexible y potente al mismo tiempo, especializado en la creación de proyectos Windows. Tiene un compilador rápido y brinda la posibilidad de obtener un ejecutable que puede ser utilizado aun cuando el programa no se encuentre instalado en la computadora.

El lenguaje tiene como conceptos clave, entre otros, la clase (*class*), que facilita la creación de tipos de datos definidos por el usuario juntamente con funciones o métodos para tratar con ellos, el encapsulamiento de datos, la asignación dinámica de memoria y la sobrecarga de operadores. Se diseñó explícitamente para lograr una mayor flexibilidad en la programación avanzada, aunque manteniendo un grado de simplicidad apreciable, basándose para todos los efectos en el lenguaje C, de ya muy



amplia difusión, y en la metodología básica del lenguaje simula, en lo referente a la programación orientada objeto. (15)

Java

La sintaxis del lenguaje heredó características de C y C++, adoptando una muy similar a la del C++. Actualmente, dentro de los lenguajes populares es uno de los mejores en cuanto a definición, debido a que goza de total independencia del implementador del lenguaje y de sus clases auxiliares. Proporciona los tipos de datos primitivos similares a los de C++, sólo que carece de punteros y mediante su biblioteca de clases estándar proporciona todas las estructuras contenedoras "clásicas" antes mencionadas. Tiene cuatro niveles de empaquetamiento: variables y funciones, al igual que los lenguajes anteriores y otros dos propios de él, denominados: clases y paquetes. Este lenguaje cuenta con una interfaz orientada a objetos para acceder de un modo portable a cualquier base de datos, promoviendo la portabilidad. Es poseedor de una extensa biblioteca utilitaria y la portabilidad alcanzada es cualitativamente superior a la que se puede obtener con los lenguajes C/C++. Java fue creado desde el inicio para ser ejecutado en cualquier clase de dispositivo o CPU, y uno de sus aspectos más interesantes es que no se compila directamente en el lenguaje máquina del CPU en uso, sino en un "pseudo lenguaje máquina" denominado "byte code", que puede ser transportado a cualquier computador en el cual se dispone de un programa especial encargado de la traducción del "byte code" al verdadero "lenguaje máquina" del CPU en uso. En otras palabras, este programa especial "interpreta" el "byte code" efectivamente, ejecutando la aplicación Java original. (16)

Este programa intérprete se conoce como "máquina virtual de Java" (JVM) o "Java Runtime Environment". Esto hace que sea más lento que un programa ya traducido al lenguaje máquina, además en las aplicaciones relativamente pequeña, los recursos de memoria que utiliza son grandes respecto a otros lenguajes. Desde sus inicios se concibió como un lenguaje muy fácil de comprender y utilizar, sin que esto signifique que su aprendizaje sea trivial. Siempre ha tenido la buena política de



estandarizar muchas bibliotecas, para una gran cantidad de aspectos que nunca fueron considerados en C ni C++, por ejemplo: la interfaz gráfica, el acceso a bases de datos, páginas Web, entre otros. El contexto de bibliotecas estandarizadas en torno a Java es tan amplio que la plataforma Java" se publicita como un conjunto de tecnologías orientadas hacia distintos tipos de aplicaciones:

- Java SE (Java Standard Edition) considera facilidades de propósito general y aplicaciones de escritorio en particular.
- Java EE (Java Enterprise Edition) para el desarrollo de aplicaciones empresariales (potencialmente sofisticadas) en servidores.
- Java ME (Java Micro Edition) dirigida a la programación de dispositivos móviles. (16)

Delphi

La plataforma de implementación (IDE) Delphi en su versión 7 está diseñada para la programación de alto nivel y de propósito general. Tiene como basamento el lenguaje Pascal. A pesar de contar con un editor de HTML, XML y Servicios y Aplicaciones Web, su principal uso está dado por la realización de aplicaciones de escritorio. Es un Software propietario. Sirve de interfaz para generar entornos gráficos, aplicaciones multimedia, Internet, gestión de datos, de sistemas, de redes, etcétera. Permite intercambiar, transformar y manipular documentos y datos XML.

En la construcción de aplicaciones nativas Windows, Delphi permite una fácil creación y reutilización de DLLs⁸, controles COM⁹, así como el desarrollo de licencias comerciales para ventas de software profesional. Construye y utiliza Windows COM, COM+, ActiveX y automatización de objetos. En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET. Existe una versión de Delphi para sistemas Unix y Linux, denominada Kylix. Es un RAD (Rapid Application Developer) y por tanto minimiza el tiempo de

⁸ **Dynamic Link Library:** Biblioteca de Enlace Dinámico

⁹ **Component Object Model:** Componente de modelo de objeto



programación mediante un ambiente de trabajo visual. Una de las principales características es su capacidad para desarrollar aplicaciones con conectividad a bases de datos de diferentes fabricantes. El programador de Delphi cuenta con una gran cantidad de componentes para realizar la conexión, manipulación, presentación y captura de los datos, algunos de ellos liberados bajo licencias de códigos abiertos o gratuitos. Estos componentes de acceso a datos pueden enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos, gracias al polimorfismo. (14)

Lenguaje C#

El lenguaje C# o también conocido como CSharp, perteneciente a la Plataforma de .NET representa el lenguaje nativo de la misma, debido a que se ha usado para escribir la mayor parte de Librerías de Clases Base del framework y está optimizado para el CLR de .NET. Sus creadores son Scott Wiltamuth y Anders Hejlsberg, este último conocido también como el creador del Lenguaje Turbo Pascal. (12)

C# es un lenguaje moderno que toma las principales características de otros como Visual Basic, C++ y Java para combinarlas. Su versión final, liberada junto con el Visual Studio Express 2013, reúne un conjunto de características que giran en torno a las comodidades y ahorro de tiempo del programador.

1.7.4 Gestores de Base de Datos

¿Qué es un Sistema Gestor de Bases de Datos?

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto. Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el



sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas. Hay muchos tipos distintos según cómo manejen los datos y muchos tamaños distintos de acuerdo a si operan en computadoras personales y con poca memoria o grandes sistemas que funcionan en *mainframes* con sistemas de almacenamiento especiales. (17)

Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información. Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero. (17)

Oracle.

Oracle es una de las mejores bases de datos que existen en el mercado, es un SGBD robusto, tiene muchas características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma. Es un producto vendible a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo sea utilizado por empresas muy grandes y multinacionales, por norma general. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de gestores de bases de datos comerciales y de la oferta de otros con licencia software libre como PostgreSQL, MySQL o FireBird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux. El principal inconveniente de utilizar Oracle como SGBD es el alto precio de su obtención. (18)

MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y



multiusuario. El objetivo que se persigue con MySQL es que cumpla con el estándar de Lenguaje de Consulta Estructurado (SQL), pero sin sacrificar velocidad, fiabilidad o usabilidad. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. En las aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a este SGBD ideal para las aplicaciones de este tipo. (19) Es un sistema de administración relacional de bases de datos, ya que archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. (19)

MySQL es software de código abierto, usa Licencia Pública General (GNU/GPL) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Usar MySQL tiene ventajas adicionales:

- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- Conectividad: Permite conexiones entre diferentes máquinas con distintos sistemas operativos.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.
- Seguridad: Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. (19)

SQLite

SQLite es un sistema de gestión de bases de datos relacional, SQLite es un proyecto de dominio público creado por D. Richard Hipp.



A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. (20)

1.7.5 Patrones Arquitectónicos

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen una escala más grande. (21)

Aunque un patrón arquitectónico comunica una imagen de un sistema, no es una arquitectura como tal. Un patrón arquitectónico es más un concepto que captura elementos esenciales de una arquitectura de software. Muchas arquitecturas diferentes pueden implementar el mismo patrón y por lo tanto compartir las mismas características. Además, los patrones son a menudo definidos como una cosa "estrictamente descrita y comúnmente disponible". Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interactuar. Cuando esto es descrito estrictamente y comúnmente disponible, es un patrón. (21)

Uno de los aspectos más importantes de los patrones arquitectónicos es que encarnan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad. A primeros de la fase de diseño, un arquitecto de software escoge qué patrones arquitectónicos mejor ofrecen las calidades deseadas para el sistema. (21)

1.7.5.1 Arquitectura en capas

Este tipo de patrón arquitectónico define como organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo que quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son consistentes de ningún detalle o interfaz de las superiores. (22)

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios, de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. (22)



Fig. 1.1: Arquitectura en Capa

Capa de presentación: es la que ve el usuario (también se la denomina "*capa de usuario*"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio. (22)



Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación. (22)

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (22)

Todas estas capas pueden residir en un único ordenador (*no es lo típico*), si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación. Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio. (22)

1.7.5.2 Modelo Cliente – Servidor de 2 capas

Las aplicaciones cliente-servidor clásicas o de 2 capas, como su nombre lo indica, agrupan la lógica de presentación (interfaz) y la lógica de aplicación en la máquina cliente y acceden a fuentes de datos compartidos a través de una conexión de red que se encuentran en el servidor de datos. (21)

La ventaja que presenta este tipo de aplicaciones es que los datos están centralizados. Esta centralización beneficia a la empresa pues es más fácil compartir los datos, se simplifica la generación de reportes y se proporciona consistencia en el acceso a los datos. (21)

A continuación citamos las principales desventajas que presentan este tipo de aplicaciones:

- **Difíciles de mantener:** Esto viene dado por el hecho de que son difíciles de mantener las reglas de negocio de la lógica de aplicación ya que estas están programadas en cada cliente y esto implica que cualquier cambio tiene que ser redistribuido en todos los clientes.
- **Se compromete la confidencialidad:** Al tener programada la lógica de aplicación en el cliente este tiene a su disposición todas las reglas de negocio de la empresa.
- **Están estrechamente limitadas a una fuente de datos:** Los clientes casi siempre están configurados para acceder a una base de datos en particular por lo que mover los datos a una base de datos diferente se hace realmente complicado. (21)



Fig. 1.2 Elementos de la arquitectura en dos capas

1.7.5.3 Modelo - Vista - Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. (21)

Descripción

- Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista. (21)

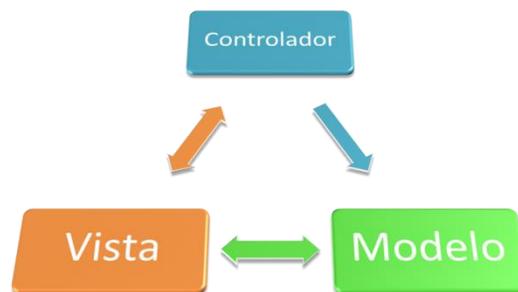


Fig. 1.3 Patrón Arquitectura Modelo Vista Controlador

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí. (21)



1.7.5.4 Arquitectura Propuesta para el desarrollo de la aplicación

Se eligió la Arquitectura en capas por la independencia que existe entre la lógica de diseño y la lógica de negocio. A demás de las ventajas que nos brinda de poder tener todas las capas en una sola Computadora.

1.7.6 Rational Rose Enterprise.

Es la herramienta CASE que comercializan los desarrolladores de UML. Propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Esta herramienta es un generador de código porque se puede generar código en distintos lenguajes de programación a partir de un diseño en UML y permite realizar la ingeniería inversa porque a partir del código de un programa, se puede obtener información sobre su diseño. Utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos. (23)

1.7.7 Embarcadero ER/Studio

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos.

Ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de



los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes. (24)

1.7.8 Fundamentación de la selección de las Herramientas

1.7.8.1 ¿Por qué seleccionar C#?

Porque reúne un conjunto de características que giran en torno a las comodidades y ahorro de tiempo del programador, entre las cuales se encuentran:

- Tipos de dato dinámico: Representa uno de los cambios más importantes, pues permite un alto grado de flexibilidad y evita el uso de las operaciones de casting¹⁰.
- Parámetros opcionales: Con esta nueva característica es posible definir un tipo de dato predeterminado en los parámetros de un método para evitar las conversiones de tipos. (11)
- Recolección de basura: Todo lenguaje incluido en la plataforma .NET tiene a su disposición el recolector de basura del CLR. Esto implica que no es necesario incluir instrucciones de destrucción de objetos en el lenguaje. (11)

1.7.8.2 ¿Por qué seleccionar SQLite?

Se utilizó como Gestor de Base de Datos SQLite por las características siguientes

- Tamaño: SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- Rendimiento de base de datos: SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- Portabilidad: SQLite se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- Estabilidad: SQLite es compatible con ACID¹¹.

¹⁰ **Casting:** Conversión de un tipo de dato a otro.



- Costo: SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente. (20)

1.9 Metodología de desarrollo de software

Una metodología de desarrollo de software es una colección de documentación formal sobre los procesos, políticas y procedimientos que intervienen en el desarrollo del software, encaminada a garantizar la eficacia, mediante el cumplimiento de los requisitos, y la eficiencia, mediante la optimización del tiempo. (25)

1.9.1 Metodologías Ágiles

Están orientadas a proyectos pequeños, constituyendo una elevada simplificación de los procesos de desarrollo de software tradicionales, sin renunciar a las prácticas esenciales que aseguran la calidad del producto obtenido (25).

Son más adecuadas para muchos de los proyectos actuales que requieren una reducción drástica de los tiempos de desarrollo, debido a que el contexto de los mismos puede ser muy cambiante.

Estas metodologías se centran en el individuo, la colaboración con el cliente y el desarrollo incremental del software mediante iteraciones muy cortas. (26)

Extreme Programming (XP)

La Programación Extrema es una metodología ligera de desarrollo de aplicaciones informáticas que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Es la más notoria de las técnicas ágiles de desarrollo de software y como todas ellas, agrega más importancia a la adaptabilidad que a la previsibilidad. (26)

Los favorecedores de XP advierten que las variaciones de los requisitos sobre el curso del ciclo de vida de la construcción son una propiedad normal, necesaria e incluso

¹¹**ACID**: Reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.



preferible del desarrollo de proyectos. Creen que ser capaz de ajustarse a los cambios de requisitos en cualquier punto de la vida del plan es un acercamiento superior y más objetivo que pretender precisar todas las necesidades al inicio del proyecto y emplear energías posteriormente en vigilar los cambios en los requisitos. (26)

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores. (26)

Scrum

Scrum es un marco de trabajo para el desarrollo de aplicaciones basado en un proceso iterativo e incremental, que es manejado usualmente en ambientes instituidos en el desarrollo ligero de software. Es un patrón de referencia que precisa un cúmulo de prácticas y roles, y que puede tomarse como partida para precisar el transcurso de desarrollo que se ejecutará durante un proyecto. Los roles primordiales en Scrum son:

- *ScrumMaster*, que mantiene los procesos y trabaja de manera análoga al director de proyecto.
- *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos)
- *Team* que contiene a los desarrolladores.

Scrum permite la creación de equipos auto-organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Scrum define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas. (27)



SXP

Según autores cubanos *SXP* es un híbrido cubano de metodologías ligeras que tiene como fundamentos las metodologías *Scrum* y *eXtreme Programming*. Está compuesta por cuatro fases principales, y entre todas se despliegan un conjunto de siete flujos de trabajo:

- *Planificación-Definición*: Se establece la visión, se fijan las expectativas y se asegura el financiamiento del proyecto
 - *Concepción inicial.*
 - *Captura de requisitos.*
 - *Diseño de metáforas.*
- *Desarrollo*: Se realiza la implementación del sistema hasta que esté listo para entregarse.
 - *Implementación.*
 - *Pruebas.*
- *Entrega*: Es la puesta en marcha del producto.
 - *Entrega de la documentación.*
- *Mantenimiento*: Se realiza el soporte para el cliente.
 - *Soporte e investigación.*

De estos flujos se ejecutan muchas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, descripción de las Historias de Usuario, diseño, implementación, organización de las iteraciones y las actividades que se efectúan para conseguir el producto, pruebas; conjuntamente con los trabajos ineludibles para cometer las pesquisas para documentar todo el proceso.

Estos flujos y actividades conciben artefactos para la documentación del expediente del proyecto. *SXP* robustece la labor en equipo y lo orienta en una misma trayectoria, con un



propósito manifiesto y además admite acompañar transparentemente el progreso de las tareas que se consuman. (28)

1.9.2 Metodología propuesta para el desarrollo de la aplicación.

En este trabajo, se decide utilizar la metodología XP ya que es ideal para grupos de desarrollo relativamente pequeños y donde el tiempo disponible para la entrega del proyecto es corto; como es el caso de la presente investigación. Además de ser de fácil realización y requiere poco papeleo por lo que se hace más cómoda su utilización. Actualmente XP es el método ágil más documentado (hay una colección de libros con XP Series de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías. (26)

Ventajas de XP

- Apropiado para entornos volátiles.
- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas) o en forma gradual.
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo.
- Estar preparados para el cambio, significa reducir su coste.
- Permite definir en cada iteración cuáles son los objetivos de la siguiente.
- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Permite la retroalimentación.
- Actualmente es la metodología ágil más extendida y documentada. Se realizan pruebas constantemente del sistema. (26)



Conclusiones

En este capítulo se estudiaron, definieron y analizaron los conceptos básicos que tienen relación con el diseño teórico planteado en la introducción, además se realizó un estudio del estado del arte a nivel internacional y nacional sobre los sistemas similares al que se desarrolla en este trabajo. También se describieron las herramientas, arquitectura, lenguajes, tecnologías y metodología utilizados en la elaboración del sistema.



CAPÍTULO 2: REQUISITOS.

2.1 Introducción

En este capítulo se describe los requisitos del sistema. Se muestra una lista de las funcionalidades y las características del sistema. Se hace referencia a los aspectos relacionados con la arquitectura del sistema. Además se incluyen las Historias de Usuarios arquitectónicamente significativas, el Plan de Iteraciones y las Tarjetas CRC.

2.2 Breve descripción de los procesos a informatizar

La principal funcionalidad es la de procesar información referida a la recolección de los datos por parte de los geólogos de campo y la modelación de la red de pozos en gráficas 2D y 3D, obteniendo así un mapa de isocontenido del área. Se desea que la modelación de los gráficos contenga todos los datos que los geólogos añaden, los datos de los pozos, la cantidad de muestras por cada pozo y el quimismo de cada muestra. Además de que permita la gestión de cada pozo, insertar, eliminar, modificar, asimismo con las muestras y el quimismo. A través de las gráficas se puede determinar la región del área que contiene los minerales que se desea mostrar, así se da una información exacta de los recursos minerales y de si pudo haber un error al introducir los datos.

2.3 Actores del Sistema.

Tabla 2.1 Descripción de los actores del negocio

Nombre del actor	Descripción
Geólogo	Es el encargado de introducir los datos recogidos por los geólogos de campo y crear los perfiles litológicos.

2.4 Funcionalidades del Sistema.

La Tabla 1.2 contiene la lista de los requerimientos funcionales que debe cumplir el sistema, los cuales fueron especificados por el cliente.



Tabla 2.2 Lista de Funcionalidades del Sistema

Funcionalidades del Sistema.
Gestionar Geólogos.
✚ Insertar Geólogos.
✚ Modificar Geólogos.
✚ Eliminar Geólogos.
✚ Mostrar Geólogos.
Gestionar Pozo.
✚ Insertar Pozo.
✚ Modificar Pozo.
✚ Eliminar Pozo.
✚ Mostrar Pozo.
Gestionar Muestra
✚ Insertar Muestra.
✚ Modificar Muestra.
✚ Eliminar Muestra.
✚ Mostrar Muestra.
Gestionar Quimismo
✚ Insertar Quimismo.
✚ Modificar Quimismo.
✚ Eliminar Quimismo.
✚ Mostrar Quimismo.
Graficar en 2D los Pozos
Graficar en 3D los Pozos
Crear Perfiles Litológicos
Importar
Exportar

2.5 Diagrama de caso de usos del sistema.

Es el artefacto más importante del flujo de trabajo de requerimientos, donde se definen las relaciones que existen entre los actores que interactúan con el sistema y los casos de usos, estos últimos constituyen las funcionalidades de la aplicación. En la figura 2.1 se muestra este diagrama. (29)



Fig. 2.1 Diagrama caso de uso del sistema

2.6 Características del sistema

Tabla 2.3 Lista de Características del Sistema

Usabilidad
Facilidad de uso por parte de los usuarios: El sistema será usado principalmente por mineros o geólogos que necesiten ver los modelos gráficos en 3D y 2D, los perfiles litológicos y gestionar información en la base de datos, debe presentar una interfaz amigable que permita la fácil interacción con la misma y posibilitar a los usuarios sin experiencia una rápida adaptación.
Especificación de la terminología utilizada: La aplicación debe contener un lenguaje y términos utilizados por los clientes en la rama abordada, con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.
Seguridad: Los datos del programa pueden ser guardados para así evitar pérdida de información.
Integridad: Validación de los datos para evitar estados inconsistentes o campos vacíos o nulos en la BD. Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones.
Requisitos del Hardware
La Estación de Trabajo debe de contar con 1GB mínimo de Memoria RAM



Requisitos del Software Plataforma .NET versión 4.0. Sistema Operativo Windows 7, 8, 8.1
Requisito para la documentación
Manual de usuario: La aplicación debe contar con un manual de usuario.

2.7 Historias de Usuario.

Las historias de usuarios (HU), son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas *tormenta de ideas (brain storm)* arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario. Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. (26)

En las tablas de las HU, se incluyen los siguientes campos:

- **Número:** contiene el identificador de la HU.
- **Usuario:** nombre del usuario que especifica la HU.
- **Nombre de historia:** contiene el nombre que identifica a la HU.
- **Prioridad en negocio:** tipo de prioridad de la HU (Muy Alta, Alta, Media o Baja).
- **Riesgo en desarrollo:** dificultad para el programador (Alto, Medio o Bajo).
- **Puntos estimados:** valor que describe la cantidad de semanas estimados para completar la HU.
- **Iteración asignada:** número de la iteración en la que se desarrollará la HU.



- **Programador responsable:** nombre del programador encargado de implementar la HU.
- **Descripción:** breve descripción del proceso que define la HU.
- **Observaciones:** comentarios aclaratorios relacionados con la HU.

A continuación se muestran las HU Gestionar Pozo y Gestionar Muestra. Para ver el resto de las HU referentes al sistema remítase al [ANEXO 1](#) del presente documento.

Tabla 2.4 HU Gestionar Pozo

Historia de Usuarios	
Número: No. 2	Usuario: Geólogo
Nombre: Gestionar Pozo	
Prioridad en el Negocio: Alta	Riesgo en el Desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 1
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: El usuario insertará los datos recogidos por los geólogos de campo para llenar el formulario del pozo, una vez insertados se podrá mostrar, modificar y eliminar.	
Observaciones: Confirmado con el cliente	



Tabla 2.5 HU Gestionar Muestra

Historia de Usuarios	
Número: No. 3	Usuario: Geólogo
Nombre: Gestionar Muestra	
Prioridad en el Negocio: Alta	Riesgo en el Desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 1
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: Una vez insertados los datos del pozo, el usuario insertará los datos de la muestra, una vez insertados se podrá mostrar, modificar y eliminar.	
Observaciones: Confirmado con el cliente	

2.9 Planificación de entregas

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor entre 1 y 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

2.9.1 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas



iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un *release*, lo cual debe ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en 3 iteraciones, la cual se explica de forma detallada a continuación:

Primera iteración:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1, 2, 3 y 4 las cuales hacen alusión a la gestión de los pozos, las muestras, el quimismo y los geólogos. Además se tendrá la primera versión de prueba, que incorporan todas las funcionalidades antes vistas, la cual se presentará al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Segunda Iteración:

Esta iteración tiene como finalidad desarrollar las historias de usuario 5 y 6. Las mismas brindan las funcionalidades para crear las gráficas 2D y 3D.

Tercera Iteración:

En la tercera iteración se llevaran a cabo las historias de usuario 7 – 9. Con estas se tendrán las funcionalidades para crear los perfiles litológicos, importar los datos de los pozos, las muestras, el quimismo y exportar los datos de los pozos, las muestras y el quimismo.

2.9.2 Estimación del esfuerzo por historias de usuario

Para el buen desarrollo de la aplicación se realizó una estimación para cada una de las historias de usuario identificadas, y se obtienen los resultados que se muestran a continuación:

Tabla 2.6 Estimación de esfuerzos por historias de usuario

Historias de usuarios	Puntos de estimación
Gestionar Geologo	1
Gestionar Pozo	2
Gestionar Muestra	2
Gestionar Quimismo	1
Crear Gráfica 2D	2
Crear Gráfica 3D	2
Crear Perfil Litológico	3
Importar	1.5
Exportar	1.5

2.9.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto guiado por la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevaran a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas la HU en cada una de las mismas.

Tabla 2.7 Plan de Iteraciones (en Semanas)

Iteración	Descripción de la Iteración	Orden de la HU a implementar	Duración HU	Duración total
Primera	Se implementarán las historias de usuarios priorizadas como muy altas relacionadas con la gestión de Pozo, gestión de Muestra y gestión de quimismo.	1. Gestionar Geólogo	1	6
		2. Gestionar Pozo	2	
		3. Gestionar Muestra	2	
		4. Gestionar Quimismo	1	



Segunda	Se implementarán las historias de usuarios priorizadas como media relacionadas con los gráficos 2D y 3D	5. Crear Gráfica 2D	2	4
		6. Crear Gráfica 3D	2	
Tercera	Se implementarán las historias de usuarios priorizadas como bajas, relacionadas con la crear perfil litológico, importar y exportar	7. Crear Perfil Litológico	3	6
		8. Importar	1.5	
		9. Exportar	1.5	
Total			16	16

2.9 Diseño

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como tarjetas CRC, las cuales se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño.

2.9.1 Tarjetas CRC

Las tarjetas CRC (siglas en inglés de Class-Responsibility-Collaborator/ Clase-Responsabilidad-Colaborador) se realizan con el objetivo de generar jerarquías de generalización/especificación o jerarquías de agregación entre las clases, permiten identificar clases y sus responsabilidades y se hacen principalmente para realizar un diseño simple y evitar que se implementen funcionalidades que no son necesarias en el producto que se desea obtener. (30)

En las tarjetas CRC se incluyen los siguientes campos:

- **Nombre de la clase:** nombre que identifica cada clase.
- **Responsabilidades:** métodos de la clase.
- **Colaborador:** clases que colaboran con cada método.



A continuación se muestran las Tarjetas CRC de las clases Gestionar Pozo y Gestionar Muestra. Para ver el resto de las tarjetas CRC referentes al sistema remítase al [Anexo 2 Tarjetas CRC](#) del presente documento.

Tabla 2.8 Tarjeta CRC del caso de uso Gestionar Pozo

Usuario: Gestionar Pozo	
Descripción: Almacena los datos del pozo	
Atributos:	
Nombre	Descripción
ID	Identificador del Pozo
BLOQUE	Valor del área del pozo
POZO	Valor del pozo
XLOGIN	Coordenada del login
YLOGIN	Coordenada del login
ZLOGIN	Coordenada del login
XAMARRE	Coordenada del de el amarre
YAMARRE	Coordenada del de el amarre
ZAMARRE	Coordenada del de el amarre
XREPLANTEO	Coordenada del de el replanteo
YREPLANTEO	Coordenada del de el replanteo
ZREPLANTEO	Coordenada del de el replanteo
DOCUMENTO	Quien documentó los datos del pozo
PERFORO	El que perforó el pozo
REVISO	El que revisó todos los datos
FECHA_INICIO	Fecha de inicio de la perforación en el



	área
FECHA_FIN	Fecha final de la perforación
AREA_AFECTADA	Área afectada por las perforaciones
NIVEL_AGUA	Nivel de agua que existe en el área
DESCRIP	Breve descripción que hace el geólogo
Responsabilidades:	
Nombre	Colaborador
insertarPozo()	POZO, conexión
modificarPozo()	POZO, conexión
eliminarPozo()	POZO, conexión
insertarMuestra()	MUESTRA, conexión
modificarMuestra()	MUESTRA, conexión
eliminarMuestra()	MUESTRA, conexión
insertarQuimismo()	QUIMISMO, conexión
modificarQuimismo()	QUIMISMO, conexión
eliminarQuimismo()	QUIMISMO, conexión
guardarBD()	POZO, QUIMISMO, MUESTRA, conexión
Importarpozo()	POZO, conexión
Importarcoordenadas()	POZO, conexión
Importarmuestra()	MUESTRA, conexión
Importarquimismo()	QUIMISMO, conexión
exportarpozo()	POZO, conexión
exportarmuestra()	MUESTRA, conexión
exportarquimismo()	QUIMISMO, conexión



Conclusiones

Como resultado de este capítulo se abordó la fase de planeación y diseño donde se delinearon las HU con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Se presentaron además las principales clases que se utilizarán en el desarrollo de la aplicación a través de las llamadas tarjetas CRC culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.



CAPITULO 3 DESCRIPCIÓN Y PRUEBA DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

Luego de concluido el capítulo anterior en el que se abordaron la fase de Planeación y Diseño, damos inicio en el presente capítulo conforme a la metodología utilizada a las fases de Desarrollo y Pruebas, en el cual se describen cada una de las tareas confeccionadas para cumplir con el desarrollo de las historias de usuarios definidas. Además se realizan las pruebas de aceptación para comprobar el correcto funcionamiento de la aplicación.

3.2 Diagrama de clases del diseño.

Este describe las clases que componen el sistema con mayor detalle. Incluye los atributos particulares de cada clase y las relaciones existentes entre ellas. Se crean para cumplimentar los requisitos funcionales y no funcionales determinados en el capítulo 2.

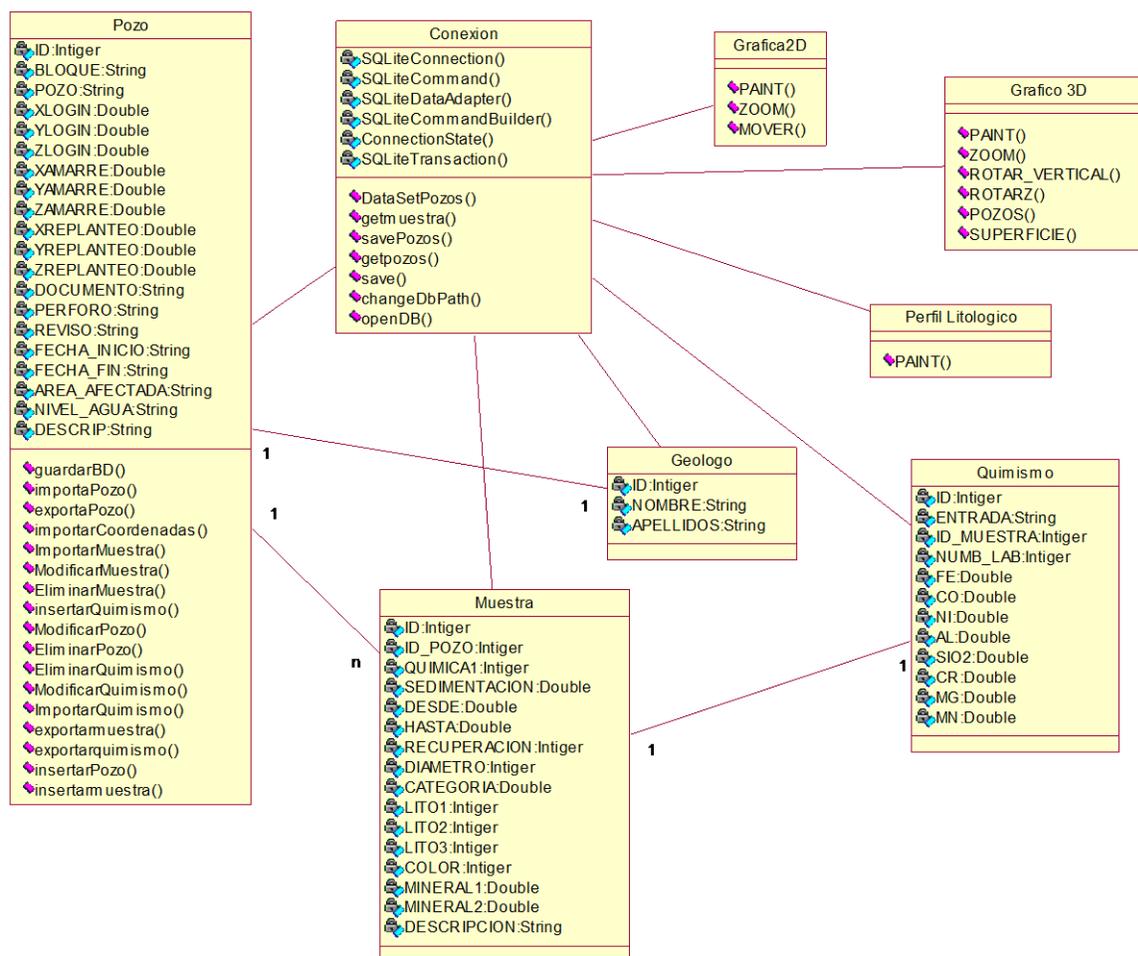


Fig. 3.1 Diagrama de clase de Diseño

Al realizarse el diagrama de clase de diseño se llegó a la conclusión de que el sistema cuenta con 4 clases fundamentales Pozo, Quimismo, Muestra, Geólogos y 3 clases para hacer la graficación de los planos en 2D, 3D y crear los perfiles litológicos

3.3 Modelo de Datos

Es un lenguaje orientado a describir una Base de Datos (BD). Típicamente un modelo de datos permite describir:

- Las estructuras de la BD: El tipo de datos que hay en la BD y la forma en que se relacionan.



- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la información deseada.

Operaciones de manipulación de los datos: Operaciones de agregado, borrado, modificación y recuperación de los datos de la BD a continuación mostramos el modelo de datos utilizado en el sistema.

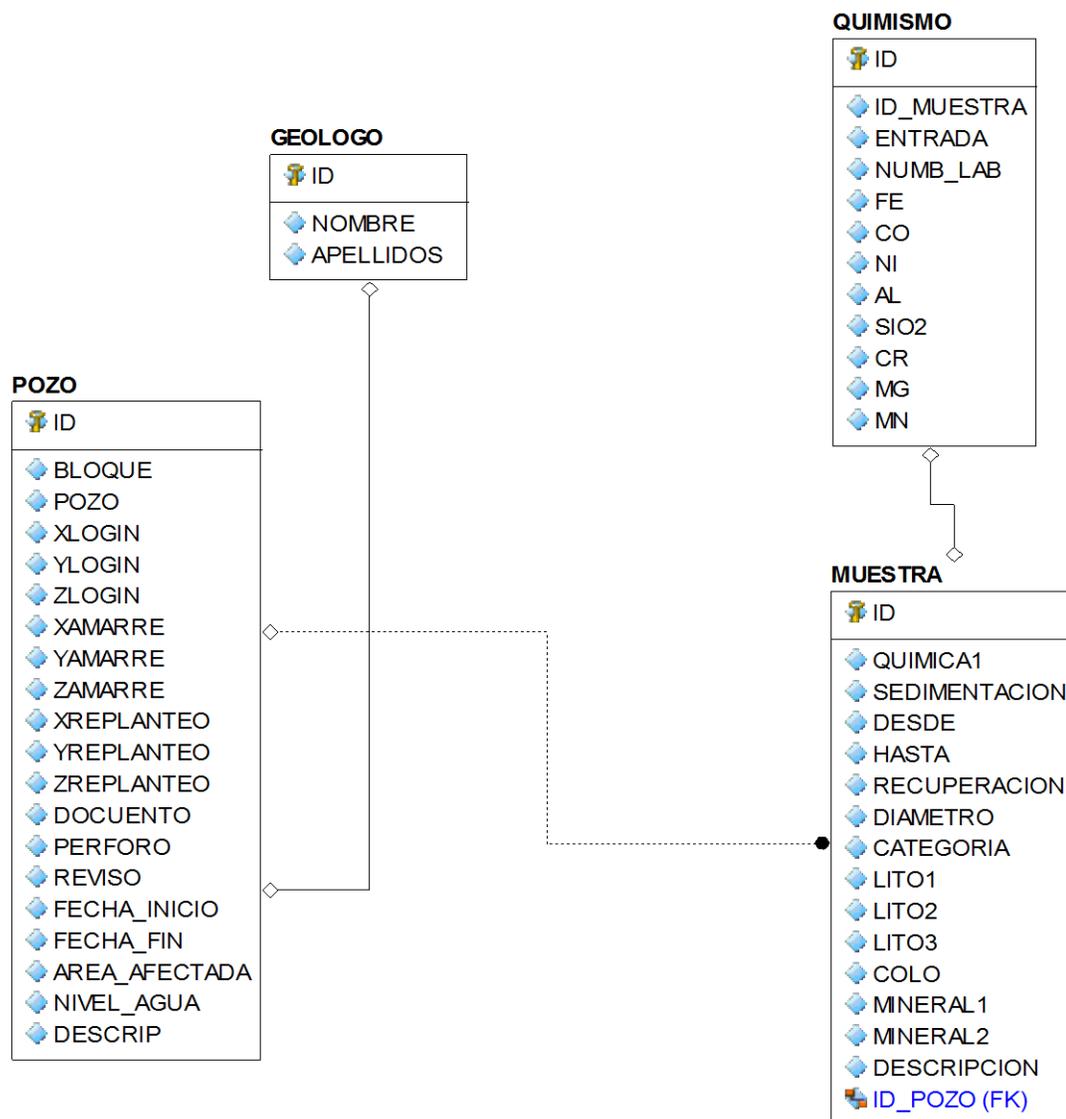


Fig. 3.2 Modelo de Datos

3.4 Interfaz de usuario.

Respondiendo a las solicitudes del cliente se escogen colores tenues para el diseño del software, se emplean palabras familiarizadas con el entorno del cliente y se diseña una interfaz sencilla, previendo que el usuario no se pierda en la navegación del software.

Fig. 3.3 Interfaz para el caso de uso Gestionar Pozo

En esta interfaz el geólogo podrá insertar o modifica los datos del pozo

Fig. 3.4 Interfaz del caso de uso Gestionar Muestra

Una vez introducido el pozo, se podrá insertar o modificar los datos de la muestra

Fig.3.5 Interfaz del caso de uso Gestionar Quimismo

Una vez insertado la muestra en la Base de Datos podemos insertar el quimismo de la muestra.

Fig. 3.6 Interfaz Principal

En esta interfaz se podrá insertar, eliminar o modificar el pozo, la muestra o el quimismo



3.5 Ayuda

El sistema contará con un manual de ayuda para el usuario, en el que se explica detalladamente cómo se trabaja con la aplicación.

3.6 Tratamientos de errores.

Para que un sistema sea confiable, este debe ser capaz de detectar la mayor cantidad de errores que sea posible y explicarle al usuario dónde es que se ha cometido el error. Los errores más frecuentes cometidos por los usuarios son: al insertar un elemento que ha sido insertado anteriormente, cuando van a eliminar o modificar uno que no está registrado en la base de datos o al realizar una de las acciones anteriores se dejan campos vacíos. Para estos, el sistema utiliza las excepciones try.....catch dándole posibilidades al usuario de rectificar su falta. Los mensajes mostrados a los usuarios son breves y orientadores, ayudando en alguna medida a estos a rectificar las causas que le dieron origen.

3.7 Diagramas de Secuencias

El diagrama de secuencia es un artefacto de UML que muestra como los objetos se comunican unos con otros para llenar los requerimientos del sistema. Da una visión gráfica de las interacciones de los actores y las operaciones del sistema a que dan origen. El diagrama de secuencia da una vista del comportamiento del sistema mostrando qué hace ante el medio y sin explicar cómo lo hace. (29)

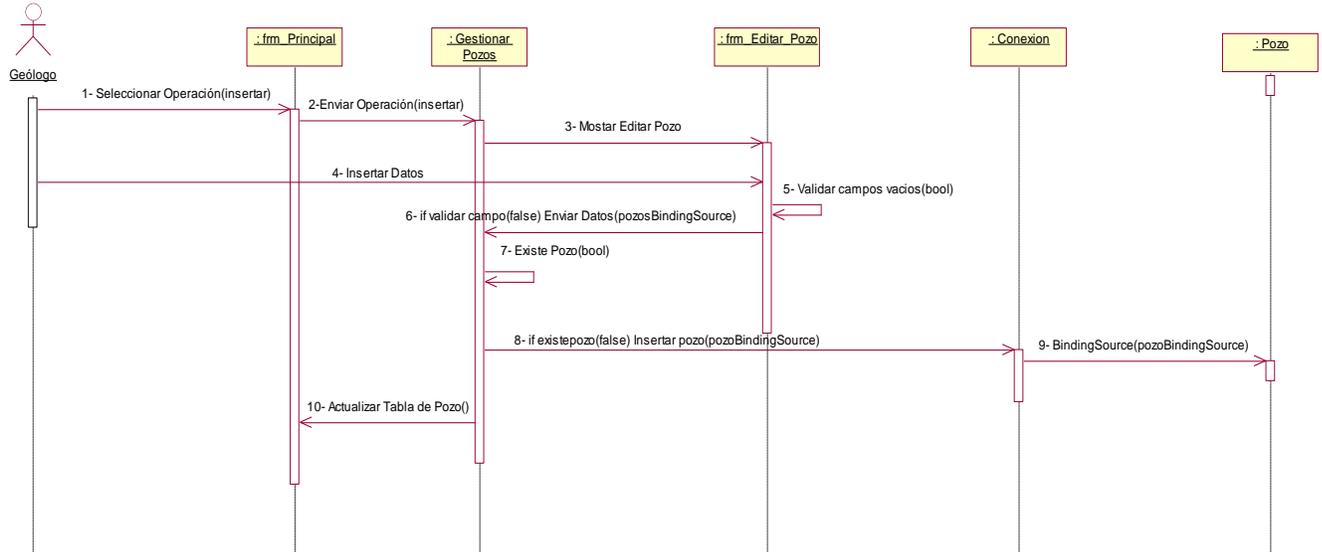


Fig. 3.7 Diagrama de secuencia Insertar Pozos

En este diagrama se muestra los eventos que se realiza entre el actor externo (Geólogo) y los eventos internos del sistema (frm_Principal, Gestionar Pozo, etc.), con los cuales se realizan los pasos necesarios para insertar un pozo.

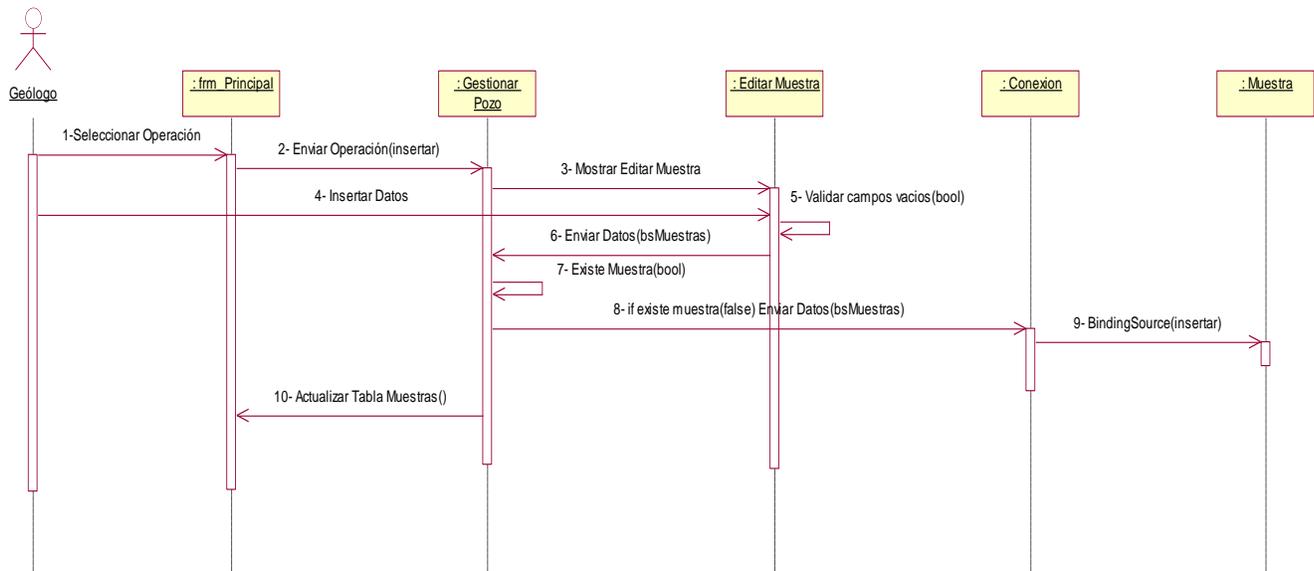


Fig. 3.8 Diagrama de secuencia Insertar Muestra

Los demás diagramas se encuentran en el [Anexo 4](#), en los diferentes diagramas aparece la clase Gestionar Pozos esto ocurre porque en el programa se decidió utilizar una clase controladora (Gestionar Pozo) y así quedaba centralizado el trabajo

3.8 Diagrama de Componente

El Diagrama de Componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. El Diagrama de Componentes modela componentes del sistema, a veces agrupados por paquetes, y las dependencias que existen entre componentes. (29)

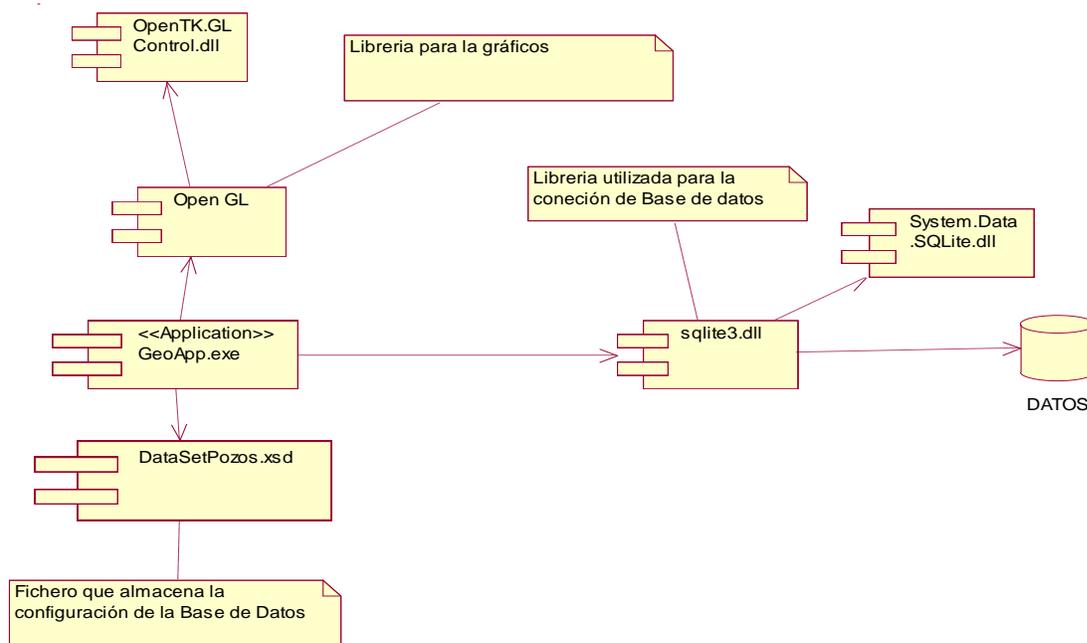


Fig. 3.9 Diagrama de Componente



En la fig.1.9 se representa las interfaces necesarias para el funcionamiento del software. La Aplicación depende de los componentes Open GI que no es más que la librería que utiliza para modelar los gráficos en 2D y 3D. Depende de la librería SQLites3.dll que es la encargada de gestionar los diferentes comandos de conexión y del archivo DataSetPozos.xsd en la cual está contenido la configuración de la Base de Datos.

3.9 Tareas de ingeniería

En XP generalmente cada historia se divide en tareas de ingeniería o tareas de programación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se crean para obtener una mejor planificación y cumplir con las funcionalidades básicas que luego conformaran las funcionalidades generales de cada historia de usuario. Las otras tareas de Ingeniería se encuentran en el [Anexo 4 Tarjetas de Ingeniería](#)

Tabla 3.1 Tarea de programación Insertar Pozo

Tarea de Programación	
No. Tarea: 4	No. Historia: 2
Nombre de la tarea: Insertar Pozo.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/1/2015	Fecha fin: 27/01/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un nuevo registro en la tabla Pozos.	

3.9.1 Pruebas

Una de las características de la metodología XP es el proceso de pruebas. Al realizar pruebas al software tanto como sea posible se aumenta la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. (26)

3.9.1.1 Pruebas de Aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que la HU ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los arquetipos anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación. (31)

Tabla 3.2 Prueba de Aceptación para la HU Gestionar Pozo

Prueba de Aceptación
HU: #1 Gestionar Pozo
Nombre: Prueba para comprobar la entrada de datos del pozo.
Descripción : Validación de los datos entrados en el pozo .
Condiciones de ejecución: El geólogo debe de introducir los datos del pozo.
Entrada/Pasos ejecución: El geólogo introduce los datos del pozo.
Resultado: Se crea un nuevo registro en la tabla pozo.
Evolución de la prueba: Aceptada

Las demás pruebas realizadas se encuentran en el [Anexo 3](#)



Conclusiones

Durante este capítulo se elaboraron las diferentes tareas de ingeniería a desarrollar por cada historia de usuario para un total de 9, las cuales sirvieron para organizar y definir el orden lógico de pasos o tareas a implementar, para dar cumplimiento a cada una de las funcionalidades definidas por el usuario. También se realizaron las pruebas de aceptación para verificar una vez terminado el producto si cumple o no con las especificaciones trazadas.

CAPÍTULO 4- ESTUDIO DE FACTIBILIDAD

4

4.1. Introducción.

En la actualidad para un desarrollo satisfactorio de cualquier proyecto se hace imprescindible el estudio de factibilidad para tener en cuenta una estimación de los costos a incurrir logrando así definir si será factible o no desarrollar el mismo.

Hay muchas formas de calcular el costo, para nuestro proyecto se utilizará la metodología Costo - Beneficio, la cual sugiere que la conveniencia de la ejecución de un proyecto se determina por la observación de ciertos factores en conjunto, estos son:

- ✚ El costo que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware / software y los costos de operación asociados.
- ✚ La efectividad que se entiende como capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo por el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad del cumplimiento del objetivo). (32)

4.2. Efectos Económicos.

Los efectos económicos se clasifican en:

- ✚ Efectos directos.
- ✚ Efectos indirectos.
- ✚ Efectos externos.
- ✚ Intangibles.

Efectos directos:

Positivos:



- ✚ Se agilizará el proceso de graficación de las áreas de perforación.
- ✚ La creación de perfiles litológicos se hará más sencilla.
- ✚ Se evitará el deterioro de las bibliografías existentes utilizadas en la realización del proceso.

Negativos:

Para usar la aplicación es imprescindible el uso de un ordenador por lo que esto trae aparejado gastos por consumo de corriente eléctrica.

Efectos indirectos:

Debido a que la aplicación no está construida con un objetivo de venta, sus efectos económicos indirectos no son de un valor significativo.

Efectos externos:

Se obtendrá un producto disponible que les facilitará todo el trabajo a los geólogos para la inserción de los datos y la graficación de las áreas de perforación.

Intangibles:

A la hora de realizar la valoración económica siempre hay elementos apreciables por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.

Con el fin de evaluar con precisión los efectos, deberán considerarse dos situaciones: la situación sin proyecto y la situación con proyecto.

Situación sin proyecto:

El desarrollo del proceso de la representación gráfica de una red de pozos, provoca una gran pérdida de tiempo a los geólogos, gastos de papel y deterioro de las bibliografías utilizadas en dichos procesos.

Situación con proyecto:

Con la realización de la aplicación se agilizó en gran manera el proceso de la representación gráfica de la red de pozos, se minimizó el uso de bibliografía en formato duro, así como el uso de papel.

4.3 Beneficios y Costos intangibles en el proyecto:

Costo:

- ✚ Resistencia al cambio.

Beneficios:

- ✚ Mayor rapidez a la hora de realizar el proceso.
- ✚ Facilidad al interpretar los resultados.
- ✚ Los datos se encuentran centralizados y disponibles para su uso en cualquier momento.

4.4 Fichas de Costo:

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una ficha de costo de un producto informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costo en Moneda Libremente Convertible

Tabla 4.1 Costo en Moneda Libremente Convertible

Ficha de Costo		
Costo Moneda Libremente Convertible		Precios(\$)
Costos Directos		
Comprar de equipos de cómputos		0,00
Alquiler de equipos de cómputos		0,00



Compra de licencia de software		0,00
Depreciación de equipos		16,66
Materiales directos		0,00
	Subtotal	16,66
Costos Indirectos		
Formación del personal que elabora el proyecto		0,00
Gastos en llamadas telefónicas		0,00
Gastos para mantenimientos del centro		0,00
Know how		0,00
Gastos en representación		0,00
	Subtotal	0,00
Gastos de Distribución y Ventas		
Participación en ferias o exposiciones		0,00
Gastos en transportación		0,00
Compras de materiales de propaganda		0,00
	Subtotal	0,00
	Total	16,66

Costo en Moneda Nacional

Tabla 4.2 Costo en Moneda Nacional

Ficha de Costo		
Costo Moneda Nacional		Precios(\$)
Costos Directos		
Salario del personal que laborará en el proyecto		100,00
12,5% de total de gastos por salarios se dedican		0,00



a la seguridad social		
9,09% de salario total, por conceptos de vacaciones a acumular		0,00
Gasto por consumo de energía eléctrica		5,40
Gasto en llamadas telefónicas		0,00
Gastos administrativos		0,00
	Subtotal	105,40
Costos Indirectos		
Know how		0,00
	Subtotal	0,00
	Total	105,40

El análisis de costo-beneficio se basa en un principio muy simple:

Compara los beneficios y los costos de un proyecto particular y si los primeros exceden a los segundos entrega un elemento de juicio inicial, que indica su aceptabilidad. (32)

Mientras que el análisis costo-efectividad sigue la misma lógica, compara los costos con las potencialidades de alcanzar más eficientemente los objetivos no expresables en moneda; si no en productos. Para esta técnica es imprescindible definir una variable directa que haga variar los costos. (32)

Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo, el tiempo en horas empleado para realizar la introducción de los datos recolectados por los geólogos de campo, la representación gráfica y el perfil litológico, la variable sería, complejidad de las pruebas que se desarrollan durante este proceso. (32)

Valores de la variable (solución manual):

- a. Insertar los datos recolectados (1 h).

- b. Graficar Perfil Litológico (168 h).
- c. Graficar Área de la Red de Pozos (240 h).

Valores de la variable (solución con la aplicación):

- a. Insertar los datos recolectados (3 min =0.05 h).
- b. Graficar Perfil Litológico (5 min = 0.083 h).
- c. Graficar Área de la Red de Pozos (6 min = 0.1 h).

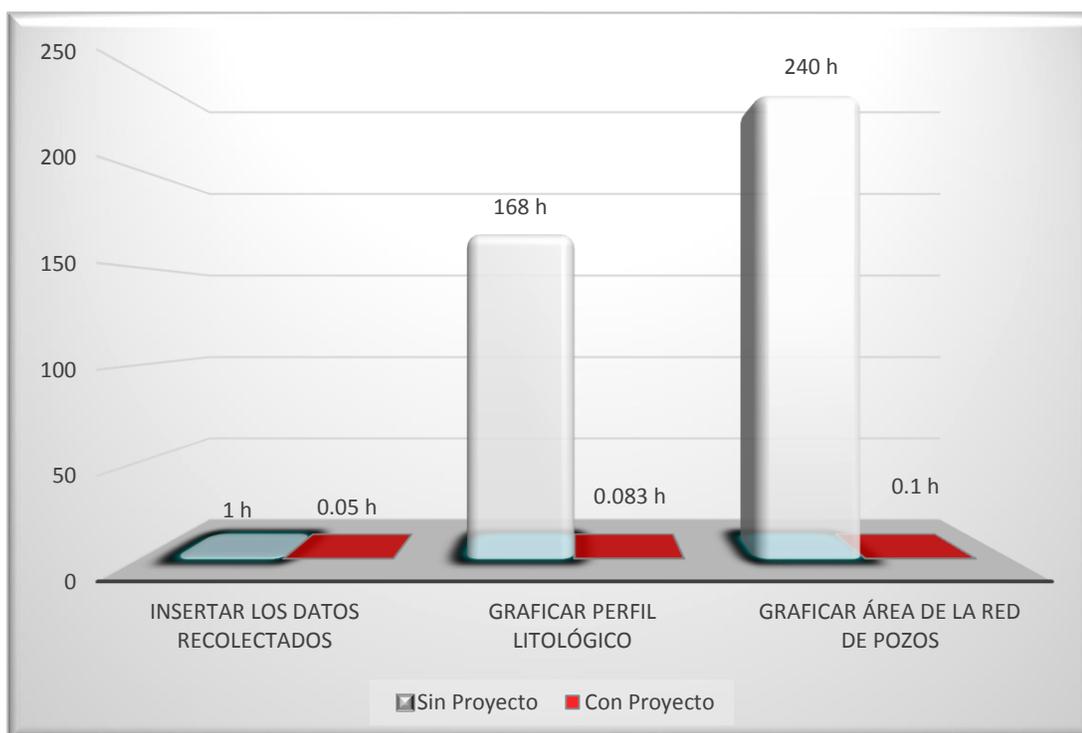


Fig. 4.1 Gráfica de comparación de la solución sin proyecto y con proyecto

El gráfico que se presentó anteriormente muestra el comportamiento de las variables, teniendo en cuenta las formas de realización de las actividades que componen el proceso. Observando los resultados reflejados en la gráfica queda demostrada la factibilidad de la aplicación desarrollada.



Conclusiones del capítulo.

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo Beneficio, se analizó los efectos económicos, los beneficios y costos intangibles, así como se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$16.66 CUC y \$105,40 CUP por mes, teniendo en cuenta que el software se desarrolló en un período de 4 meses entonces el costo total es de \$66,64 CUC y \$ 421,60 CUP, demostrándose la factibilidad económica del proyecto.



CONCLUSIONES GENERALES

Con el desarrollo del este proyecto se dio cumplimiento al objetivo general de esta investigación, arribándose a las siguientes conclusiones:

- La búsqueda y análisis de sistemas semejantes demostró, que las herramientas internacionales poseen características no requeridas por el cliente y son propietarias, mientras que a nivel nacional no se encontraron soluciones que automaticen el proceso para la representación gráfica de mapas de isocontenido y perfil litológico, evidenciándose la necesidad de crear un nuevo producto.
- Se determinó el empleo de la plataforma Microsoft .NET, Visual Studio Express 2013 como entorno de desarrollo, C# como lenguaje de programación, SQLite como Sistema de Gestión de Base de Datos; herramientas que favorecieron la realización del sistema con las características determinadas por el cliente.
- La arquitectura en capas como patrón arquitectónico.
- Se realizó la Ingeniería de Software, aplicando la metodología XP, lo que facilitó la realización del software, y a su vez demuestra las amplias potencialidades y ventajas que brinda esta metodología.
- Se realizó un estudio de la factibilidad técnica y económica, esta última mediante la evaluación costo beneficio y la confección de la ficha de costo, arrojando como resultado un costo de \$66,64 CUC y \$ 421,60 CUP, demostrándose que es factible el proyecto.
- Se desarrolló el Sistema para la representación gráfica de mapas de isocontenido y perfil litológico en la estimación de recursos minerales en el Instituto Minero Metalúrgico de Moa.

Se concluye que los objetivos planteados en el presente trabajo han sido cumplidos satisfactoriamente y se incluyen una serie de recomendaciones a tener en cuenta para versiones posteriores del producto.



RECOMENDACIONES

Continuar el desarrollo del software, para alcanzar el objetivo final, que es poder estimar los recursos minerales en un área determinada.

- Introducir los resultados de este trabajo en la carrera de Geología- Minas, para que los profesores puedan modelar los mapas de isocontenido para la estimación de recursos minerales.
- Presentar el resultado obtenido en sesiones científicas y eventos.
- Poder Agregar más funciones con respecto a la estimación de recursos minerales en Versiones posteriores.

Referencias

1. <http://www.ismm.edu.cu/universidad/historia>. [En línea] 11 de Febrero de 2015.
2. *El perfil o corte geológico. Algunos ejemplos de casos*. Obando, Tupak Ing en Geología, Doctorado y Master en Geología y Gestión Ambiental. Huelva, España : s.n., 2009.
3. BENITO, A . *Manual de Roca Ornamentales: Prospección, Explotación, Elaboración, Colocación*. 2 ed. Madrid : s.n., 1996. pág. 696.
4. <http://www.arcinfo.com>. [En línea] 11 de Febrero de 2015.
5. <http://www.mapinfo.com>. [En línea] 11 de Febrero de 2015.
6. <http://www.arcgis.com>. [En línea] 15 de Enero de 2015.
7. <http://qgis.org>. [En línea] 15 de Enero de 2015.
8. VALDÉS FURRIEL, JON. Cálculo de la triangulación de Delaunay en la GPU. 1999. <http://hombrealto.com/web/files/delaunayGPU.pdf>. [En línea] 11 de Febrero de 2015.
9. <http://asignatura.us.es/fgcitig/contenidos/gctem3ma.htm>. [En línea] 11 de Febrero de 2015.
10. <https://www.visualstudio.com/es-es/products/visual-studio-express-vs.aspx>. [En línea] 8 de Enero de 2015.
11. MSDN. MSDN. [En línea] Microsoft, Enero de 2012. [Citado el: 10 de Diciembre de 2014.] <http://www.msdn.microsoft.com/>.
12. —. MSDN. [En línea] Microsoft, 2008. [Citado el: 10 de Diciembre de 2014.] <http://msdn.microsoft.com/es-es/library/ms954595.aspx>.
13. <https://www.opengl.org>. [En línea] 12 de Febrero de 2015.
14. Marteens, L. *La cara oculta de delphi*. La Habana : Félix Varela, 2005. pág. 296.
15. *La cara Oculta de C/C++*. La Habana : Félix Varela, 2005.
16. <https://www.java.com/es>. [En línea] 11 de Febrero de 2015.
17. Matos, R.M. *Sistemas de Bases de Datos*. 2 ed. La Habana : Félix Varela, 2006. pág. 165.
18. <http://www.oracle.com/es/>. [En línea] 11 de Febrero de 2015.
19. <http://www.mysql.com>. [En línea] 11 de Febrero de 2015.
20. <https://www.sqlite.org/>. [En línea] 8 de Enero de 2015.
21. Pressman, Roger S. *Ingeniería de Softwar, un enfoque práctico*. s.l. : McGraw-Hill, 2011.
22. Ivar, Jacobson, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. México : Addison-Wesley, 1999.
23. <http://www.rational.com>. [En línea] 11 de Febrero de 2015.
24. LAITANO, P.A ROMMELL. *Clase de Ingeniería de Software Asistido por Computador*. HONDURAS : Universidad Católica de Honduras.
25. Letelier, T. P. y Sánchez, L. E. A. *Metodologías Ágiles en el Desarrollo de Software*. 2003.
26. Letelier, Patricio y Penadés, María Carmen. *Letelier, Patricio y Penadés, María Carmen. Metodologías Ágiles para el Desarrollo de Software: eXtreme Programming (XP)*. 2003.
27. Schwaber, Ken. *Agile Project Management with Scrum*. s.l. : Microsoft Press. ISBN 0-7356-1993-X.
28. García de la Puente, S. J. *SXP, metodología de desarrollo de software*. s.l. : Serie científica de la Universidad de las Ciencias Informáticas UCI , 2011. Vol. 4.



29. LARMAN, C. *UML y patrones. Introducción al análisis y diseño orientado a objeto*. 3 ed. La Habana : Félix Varela, 2004. pág. 507.
30. BECK, K. *Extreme Programming Explained. Embrace Change*. s.l. : Pearson Education, 1999.
31. Wesley, A. *Una explicación de la programación extrema. Aceptar el Cambio*. 2000.
32. GARCÍA, Dra. Ana María PÉREZ. *Procedimientos para la elaboración de la ficha de costo de un producto informático*. Villa Clara : Facultad MFC UCLV.
33. Cepero Abreu, Eberto y Valdés Pérez , Ingris. *Componente Genérico para la comunicación en sistemas distribuidos*. 2011.
34. Gerardo Antonio Cabero, Daniel Maldonado.
<http://es.scribd.com/doc/52882068/SQLite>. [En línea] 11 de Febrero de 2015.
35. <http://sig.cea.es/SIG>. [En línea] 11 de Febrero de 2015.

Bibliografía

1. <http://www.ismm.edu.cu/universidad/historia>. [En línea] 11 de Febrero de 2015.
2. *El perfil o corte geológico. Algunos ejemplos de casos*. Obando, Tupak Ing en Geología, Doctorado y Master en Geología y Gestión Ambiental. Huelva, España : s.n., 2009.
3. BENITO, A . *Manual de Roca Ornamentales: Prospección, Explotación, Elaboración, Colocación*. 2 ed. Madrid : s.n., 1996. pág. 696.
4. <http://www.arcinfo.com>. [En línea] 11 de Febrero de 2015.
5. <http://www.mapinfo.com>. [En línea] 11 de Febrero de 2015.
6. <http://www.arcgis.com>. [En línea] 15 de Enero de 2015.
7. <http://qgis.org>. [En línea] 15 de Enero de 2015.
8. VALDÉS FURRIEL, JON. Cálculo de la triangulación de Delaunay en la GPU. 1999. <http://hombrealto.com/web/files/delaunayGPU.pdf>. [En línea] 11 de Febrero de 2015.
9. <http://asignatura.us.es/fgcitig/contenidos/gctem3ma.htm>. [En línea] 11 de Febrero de 2015.
10. <https://www.visualstudio.com/es-es/products/visual-studio-express-vs.aspx>. [En línea] 8 de Enero de 2015.
11. MSDN. MSDN. [En línea] Microsoft, Enero de 2012. [Citado el: 10 de Diciembre de 2014.] <http://www.msdn.microsoft.com/>.
12. —. MSDN. [En línea] Microsoft, 2008. [Citado el: 10 de Diciembre de 2014.] <http://msdn.microsoft.com/es-es/library/ms954595.aspx>.
13. <https://www.opengl.org>. [En línea] 12 de Febrero de 2015.
14. Marteens, L. *La cara oculta de delphi*. La Habana : Félix Varela, 2005. pág. 296.
15. *La cara Oculta de C/C++*. La Habana : Félix Varela, 2005.
16. <https://www.java.com/es>. [En línea] 11 de Febrero de 2015.
17. Matos, R.M. *Sistemas de Bases de Datos*. 2 ed. La Habana : Félix Varela, 2006. pág. 165.
18. <http://www.oracle.com/es/>. [En línea] 11 de Febrero de 2015.
19. <http://www.mysql.com>. [En línea] 11 de Febrero de 2015.
20. <https://www.sqlite.org/>. [En línea] 8 de Enero de 2015.
21. Pressman, Roger S. *Ingeniería de Softwar, un enfoque práctico*. s.l. : McGraw-Hill, 2011.
22. Ivar, Jacobson, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. México : Addison-Wesley, 1999.
23. <http://www.rational.com>. [En línea] 11 de Febrero de 2015.
24. LAITANO, P.A ROMMELL. *Clase de Ingeniería de Software Asistido por Computador*. HONDURAS : Universidad Católica de Honduras.
25. Letelier, T. P. y Sánchez, L. E. A. *Metodologías Ágiles en el Desarrollo de Software*. 2003.
26. Letelier, Patricio y Penadés, María Carmen. *Letelier, Patricio y Penadés, María Carmen. Metodologías Ágiles para el Desarrollo de Software: eXtreme Programming (XP)*. 2003.
27. Schwaber, Ken. *Agile Project Management with Scrum*. s.l. : Microsoft Press. ISBN 0-7356-1993-X.
28. García de la Puente, S. J. *SXP, metodología de desarrollo de software*. s.l. : Serie científica de la Universidad de las Ciencias Informáticas UCI , 2011. Vol. 4.
29. LARMAN, C. *UML y patrones. Introducción al análisis y diseño orientado a objeto*. 3 ed. La Habana : Félix Varela, 2004. pág. 507.



30. BECK, K. *Extreme Programming Explained. Embrace Change.* s.l. : Pearson Education, 1999.
31. Wesley, A. *Una explicación de la programación extrema. Aceptar el Cambio.* 2000.
32. GARCÍA, Dra. Ana María PÉREZ. *Procedimientos para la elaboración de la ficha de costo de un producto informático.* Villa Clara : Facultad MFC UCLV.
33. Cepero Abreu, Eberto y Valdés Pérez , Ingris. *Componente Genérico para la comunicación en sistemas distribuidos.* 2011.
34. Gerardo Antonio Cabero, Daniel Maldonado.
<http://es.scribd.com/doc/52882068/SQLite>. [En línea] 11 de Febrero de 2015.
35. <http://sig.cea.es/SIG>. [En línea] 11 de Febrero de 2015.

Anexo 1 Historias de Usuario

Tabla 5.1 HU Gestionar Quimismo

Historia de Usuarios	
Número: No. 4	Usuario: Geólogo
Nombre: Gestionar Quimismo	
Prioridad en el Negocio: Alta	Riesgo en el Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 1
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: Una vez esté registrada la muestra, el usuario insertará los datos del Quimismo, una vez insertados, se podrá mostrar, modificar y eliminar.	
Observaciones: Confirmado con el cliente	

Tabla 5.2 HU Gestionar Geólogo

Historia de Usuarios	
Número: No. 1	Usuario: Geólogo
Nombre: Gestionar Geólogo	
Prioridad en el Negocio: Alta	Riesgo en el Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 1
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: El usuario insertará los datos del Geólogo, una vez insertados se podrá mostrar, modificar y eliminar.	
Observaciones: Confirmado con el cliente	

Tabla 5.3 HU Crear Gráfico 2D

Historia de Usuarios	
Número: No. 5	Usuario: Geólogo
Nombre: Crear Gráfico 2D	



Yosbani Bientz Aguilera

Prioridad en el Negocio: Media	Riesgo en el Desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: El usuario una vez ingresado los datos correspondientes a los pozos, las muestras, y el quimismo, mostrara el gráfico 2D.	
Observaciones: Confirmado con el cliente	

Tabla 5.4 HU Crear Gráfico 3D

Historia de Usuarios	
Número: No. 6	Usuario: Geólogo
Nombre: Crear Gráfico 3D	
Prioridad en el Negocio: Media	Riesgo en el Desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: : El usuario una vez ingresado los datos correspondientes a los pozos, las muestras, y el quimismo, mostrara el gráfico 3D.	
Observaciones: Confirmado con el cliente	

Tabla 5.5 HU Crear Perfil Litológicos

Historia de Usuarios	
Número: No. 7	Usuario: Geólogo
Nombre: Crear Perfil Litológico	
Prioridad en el Negocio: Baja	Riesgo en el Desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 3
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: El usuario una vez ingresado los datos correspondientes a los pozos, las muestras, y el quimismo, mostrara el gráfico en 3D y elegirá los pozos a los cuales les desea hacer el perfil litológico.	
Observaciones: Confirmado con el cliente	



Tabla 5.6 HU Importar

Historia de Usuarios	
Número: No. 8	Usuario: Geólogo
Nombre: Importar	
Prioridad en el Negocio: Baja	Riesgo en el Desarrollo: Alto
Puntos estimados: 1.5	Iteración Asignada: 3
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: El usuario tendrá la posibilidad de importar los datos de los pozos, las muestras y el quimismo desde archivos con la extensión csv.	
Observaciones: Confirmado con el cliente	

Tabla 5.7 HU Exportar

Historia de Usuarios	
Número: No. 9	Usuario: Geólogo
Nombre: Exportar	
Prioridad en el Negocio: Baja	Riesgo en el Desarrollo: Alto
Puntos estimados: 1.5	Iteración Asignada: 3
Programador Responsable: Yosbani Bientz Aguilera	
Descripción: El usuario una vez ingresado los datos correspondientes podrá exportar los datos de los pozos, las muestras y los del quimismo, en un archivo con extensión csv.	
Observaciones: Confirmado con el cliente	

Anexo 2 Tarjetas CRC

Tabla 5.8 Tarjeta CRC del caso de uso Gestionar Geólogo

Usuario: Gestionar Muestra	
Descripción: Almacena los datos de la muestra	
Atributos:	
Nombre	Descripción
ID	Identificador de la Muestra
ID_POZO	Identificador del Pozo
QUIMICA1	Valor del quimismo
SEDIMENTACION	Sedimentación de la muestra
DESDE	Desde donde empieza esa muestra
HASTA	Donde termina la muestra
RECUPERACION	% de recuperación de la muestra que se recoge
DIAMETRO	Diámetro de la muestra
CATEGORIA	Categoría de la muestra
LITO1	Valor litológico
LITO2	Valor litológico
LITO3	Valor litológico
COLOR	Color de la muestra
MINERAL1	Tipo de mineral
MINERAL2	Tipo de mineral
DESCRIPCION	Descripción de la muestra
Responsabilidades:	
Nombre	Colaborador

Tabla 5.9 Tarjeta CRC del caso de uso Gestionar Geólogo

Usuario: Gestionar Geologo	
Descripción: Almacena los datos del geólogo	
Atributos:	
Nombre	Descripción
ID	Identificador del Geólogo
NOMBRE	Nombre del Geólogo
APELLIDO	Apellidos del Geólogo
Responsabilidades:	
Nombre	Colaborador

Tabla 6.1 Tarjeta CRC del caso de uso Gestionar Quimismo

Usuario: Gestionar Quimismo	
Descripción: Almacena los datos del Quimismo de las muestras	
Atributos:	
Nombre	Descripción
ID	Identificador del Quimismo
ENTRADA	Entrada que se hace en el laboratorio
ID_MUESTRA	Identificador de la muestra que corresponde a ese quimismo
NUMB_LAB	número del laboratorio donde se obtuvo el quimismo
FE	Cantidad de hierro
CO	Cantidad de cobalto
NI	Cantidad de níquel
AL	Cantidad de aluminio
SIO2	Cantidad de sílice
CR	Cantidad de cromo



MG	Cantidad de magnesio
MN	Cantidad de manganeso
Responsabilidades:	
Nombre	Colaborador

Tabla 6.2 Tarjeta CRC del caso de uso Crear Gráfica 2D

Usuario: Crear Grafica 2D	
Descripción: Dibuja la gráfica 2D a partir de los datos de los pozos, la muestras y el quimismo	
Atributos:	
Nombre	Descripción
Responsabilidades:	
Nombre	Colaborador
PAINT()	POZO, conexión
ZOOM()	
MOVER()	

Tabla 6.3 Tarjeta CRC del caso de uso Crear Gráfica 3D

Usuario: Crear Grafica 3D	
Descripción: Dibuja la gráfica 3D a partir de los datos de los pozos, la muestras y el quimismo	
Atributos:	
Nombre	Descripción
Responsabilidades:	
Nombre	Colaborador
PAINT()	POZO, conexión
ZOOM()	
ROTAR_VERTICAL()	



ROTAR()	
MOVER()	

Tabla 6.4 Tarjeta CRC del caso de uso Crear Perfil Litológico

Usuario: Crear Perfil Litologico	
Descripción: Dibuja la el perfil litológico correspondiente a los pozos seleccionados	
Atributos:	
Nombre	Descripción
Responsabilidades:	
Nombre	Colaborador
PAINT()	POZO, conexión

Tabla 6.5 Tarjeta CRC del caso de uso Importar

Usuario: Importar	
Descripción: Importa los datos del Pozo, la Muestra o el Quimismo según elija el Geólogo	
Atributos:	
Nombre	Descripción
Responsabilidades:	
Nombre	Colaborador
importarPozo()	POZO, conexión
importarMuestra()	POZO, conexión
importarQuimismo()	POZO, conexión

Tabla 6.6 Tarjeta CRC del caso de uso Exportar

Nombre de la Clase: Exportar
Tipo de clase: Lógica del Negocio



Responsabilidades	Geólogo
exportarPozo	POZO, conexión
exportarMuestra()	POZO, conexión
exportarQuimismo	POZO, conexión

Anexo 4 Pruebas de Aceptación

Tabla 6.7 Prueba del HU Gestionar Muestra

Prueba de Aceptación
HU: 3
Nombre: Gestionar Muestra
Descripción: El geólogo debe suministrar los datos al sistema de las muestras para luego mostrarla, modificarla y eliminarla.
Condiciones de ejecución: Validación de los datos entrados en la muestra.
Entrada/Pasos ejecución: El geólogo introduce los datos del pozo.
Resultado: Se crea un nuevo registro en la tabla Muestra.
Evolución de la prueba: Aceptada.

Tabla 6.8 Prueba del HU Gestionar Quimismo

Prueba de Aceptación
HU: 4
Nombre: Gestionar Quimismo
Descripción: El geólogo debe suministrar los datos al sistema de las muestras químicas para luego mostrarla, modificarla y eliminarla.
Condiciones de ejecución: Validación de los datos entrados en el quimismo.
Entrada/Pasos ejecución: El geólogo introduce los datos del quimismo.
Resultado: Se crea un nuevo registro en la tabla geólogo.
Evolución de la prueba: Aceptada.

Tabla 6.9 Prueba del HU Gestionar Geólogo

Prueba de Aceptación
HU: 1
Nombre: Gestionar Geólogo



Descripción: El geólogo debe suministrar los datos correspondientes al nombre y los apellidos de los geólogos de campo.
Condiciones de ejecución Validación de los datos entrados en geólogo.
Entrada/Pasos ejecución: El geólogo introduce los datos del geologo
Resultado: Se crea un nuevo registro en la tabla Geologo
Evolución de la prueba: Aceptada

Tabla 7.1 Prueba del HU Crear Gráfica 2D

Prueba de Aceptación
HU: 5
Nombre: Crear Gráfica 2D
Descripción: El geólogo con los datos suministrados, se dispondrá a crear la gráfica 2D.
Condiciones de ejecución: Los datos de los pozos, la muestra y el quimismo, deben de haber sido insertadas correctamente.
Entrada/Pasos ejecución: Crear la gráfica 2D
Resultado: La gráfica se crea correctamente
Evolución de la prueba: Aceptada

Tabla 7.2 Prueba del HU Crear Gráfica 3D

Prueba de Aceptación
HU: 6
Nombre: Crear Gráfica 3D
Descripción: El geólogo con los datos suministrados, se dispondrá a crear la gráfica 3D.
Condiciones de ejecución: Los datos de los pozos, la muestra y el quimismo, deben de haber sido insertadas correctamente.
Entrada/Pasos ejecución: Crear gráfica 3D.
Resultado: La gráfica se crea correctamente.
Evolución de la prueba: Aceptada



Tabla 7.3 Prueba del HU Crear Perfil Litológico

Prueba de Aceptación
HU: 7
Nombre: Crear Perfil Litologico
Descripción: Una vez creado la gráfica 3D el geólogo podrá elegir los pozos de los cuales quiere crear el perfil litológico.
Condiciones de ejecución: La gráfica 3D debe de haber sido creada.
Entrada/Pasos ejecución: Crear perfil litológico.
Resultado: El perfil se crea correctamente.
Evolución de la prueba: Aceptada

Tabla 7.4 Prueba del HU Importar

Prueba de Aceptación
HU: 8
Nombre: Importar
Descripción: El geólogo tendrá la oportunidad de importar datos correspondientes a los pozos, las muestras y el quimismo desde archivos con extensión csv.
Condiciones de ejecución: Debe de contar con los datos con esa extensión (csv).
Entrada/Pasos ejecución: Seleccionar que tipo de datos desea importar(Pozo, Muestra, Quimismo).
Resultado: Se importa correctamente los tipos datos seleccionados (Pozo, Muestra, Quimismo).
Evolución de la prueba: Aceptada

Tabla 7.5 Prueba del HU Exportar

Prueba de Aceptación
HU: 9



Nombre: Exportar
Descripción: Una vez el geólogo haya introducido todos los datos correspondiente a los pozos, a las muestras y al quimismo podrá exportar cada uno de los datos en archivos con extensión csv.
Condiciones de ejecución: Deben de haber sido introducido los datos correctamente (Pozo, Muestra, Quimismo).
Entrada/Pasos ejecución: Seleccionar que tipo de datos desea exportar(Pozo, Muestra, Quimismo).
Resultado: Se exporta correctamente los tipos de datos seleccionado (Pozo, Muestra, Quimismo).
Evolución de la prueba: Aceptada



Anexo 4 Tarjetas de Ingeniería

Tabla 7.6 Tarea de Ingeniería Insertar Geologo

Tarea de Programación	
No. Tarea: 1	No. Historia: 1
Nombre de la tarea: Insertar Geologo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 5/1/2015	Fecha fin: 12/01/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un nuevo registro en la tabla Geologo.	

Tabla 7.7 Tarea de Ingeniería Modificar Geologo

Tarea de Programación	
No. Tarea: 2	No. Historia: 1
Nombre de la tarea: Modificar Geologo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 5/1/2015	Fecha fin: 12/01/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite modificar un registro en la tabla Geologo.	

Tabla 7.8 Tarea de Ingeniería Eliminar Geologo

Tarea de Programación	
No. Tarea: 3	No. Historia: 1
Nombre de la tarea: Eliminar Geologo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 5/1/2015	Fecha fin: 12/01/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite eliminar un registro en la tabla Geologo.	

Tabla 7.9 Tarea de Ingeniería Modificar Pozo

Tarea de Programación	
No. Tarea: 5	No. Historia: 2
Nombre de la tarea: Modificar Pozo.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/1/2015	Fecha fin: 27/01/2015



Yosbani Bientz Aguilera

Programador responsable: Yosbani Bientz Aguilera
Descripción: Permite modificar un registro en la tabla Pozo.

Tabla 8.1 Tarea de Ingeniería Eliminar Pozo

Tarea de Programación	
No. Tarea: 6	No. Historia: 2
Nombre de la tarea: Eliminar Pozo.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/1/2015	Fecha fin: 27/01/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite eliminar un registro en la tabla Pozo.	

Tabla 8.2 Tarea de Ingeniería Insertar Muestra

Tarea de Programación	
No. Tarea: 7	No. Historia: 3
Nombre de la tarea: Insertar Muestra.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28/1/2015	Fecha fin: 11/02/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un nuevo registro en la tabla Muestra.	



Tabla 8.3 Tarea de Ingeniería Modificar Muestra

Tarea de Programación	
No. Tarea: 8	No. Historia: 2
Nombre de la tarea: Modificar Muestra.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28/1/2015	Fecha fin: 11/02/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite modificar un registro en la tabla Muestra.	

Tabla 8.4 Tarea de Ingeniería Eliminar Muestra

Tarea de Programación	
No. Tarea: 9	No. Historia: 2
Nombre de la tarea: Eliminar Muestra.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28/1/2015	Fecha fin: 11/02/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite eliminar un registro en la tabla Muestra.	

Tabla 8.5 Tarea de Ingeniería Insertar Quimismo

Tarea de Programación	
No. Tarea: 10	No. Historia: 4
Nombre de la tarea: Insertar Quimismo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/02/2015	Fecha fin: 19/02/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un nuevo registro en la tabla Quimismo.	

Tabla 8.6 Tarea de Ingeniería Modificar Quimismo

Tarea de Programación	
No. Tarea: 11	No. Historia: 4
Nombre de la tarea: Modificar Quimismo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/02/2015	Fecha fin: 19/02/2015



Yosbani Bientz Aguilera

Programador responsable: Yosbani Bientz Aguilera
Descripción: Permite modificar un registro en la tabla Quimismo.

Tabla 8.7 Tarea de Ingeniería Eliminar Quimismo

Tarea de Programación	
No. Tarea: 12	No. Historia: 4
Nombre de la tarea: Eliminar Quimismo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/02/2015	Fecha fin: 19/02/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite eliminar un registro en la tabla Quimismo.	

Tabla 8.8 Tarea de Ingeniería Crear Grafica 2D

Tarea de Programación	
No. Tarea: 13	No. Historia: 5
Nombre de la tarea: Crear Grafica 2D.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/02/2015	Fecha fin: 7/03/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un Gráfica 2D.	

Tabla 8.9 Tarea de Ingeniería Crear Grafica 3D

Tarea de Programación	
No. Tarea: 14	No. Historia: 6
Nombre de la tarea: Crear Grafica 3D.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 7/03/2015	Fecha fin: 21/03/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un Gráfica 3D.	

Tabla 9.1 Tarea de Ingeniería Crear Perfil Litologico

Tarea de Programación	
No. Tarea: 15	No. Historia: 7



Yosbani Bientz Aguilera

Nombre de la tarea: Crear Perfil Litologico.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 22/03/2015	Fecha fin: 12/04/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite la creación de un Gráfico que representa el Perfil Litológico.	

Tabla 9.2 Tarea de Ingeniería Importar

Tarea de Programación	
No. Tarea: 16	No. Historia: 8
Nombre de la tarea: Importar.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha inicio: 13/04/2015	Fecha fin: 24/04/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite importar los datos de los Pozos, las Muestras y el Quimismo.	

Tabla 9.3 Tarea de Ingeniería Exportar

Tarea de Programación	
No. Tarea: 17	No. Historia: 9
Nombre de la tarea: Exportar.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha inicio: 25/04/2015	Fecha fin: 6/05/2015
Programador responsable: Yosbani Bientz Aguilera	
Descripción: Permite exportar los datos de los Pozos, las Muestras y el Quimismo.	

Anexo 5 Diagramas de Secuencia

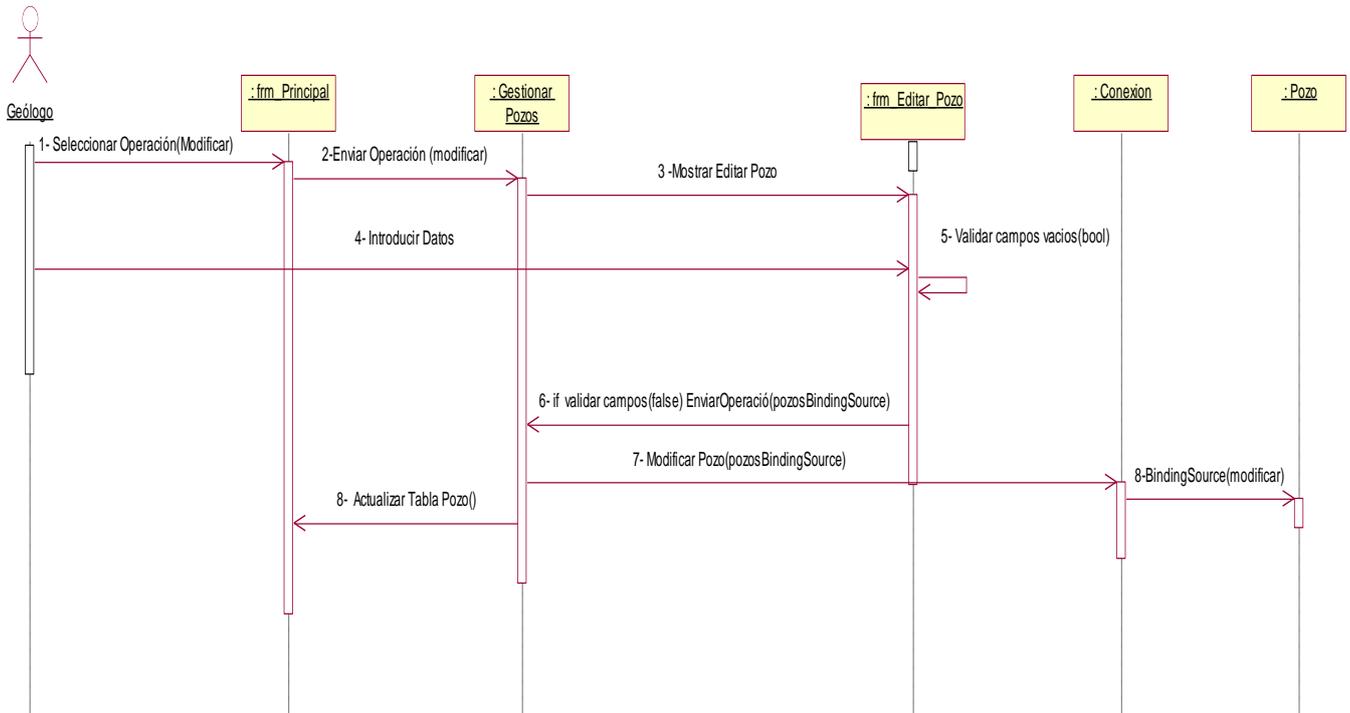


Fig. 5.1 Diagrama de Secuencia Modificar Pozo

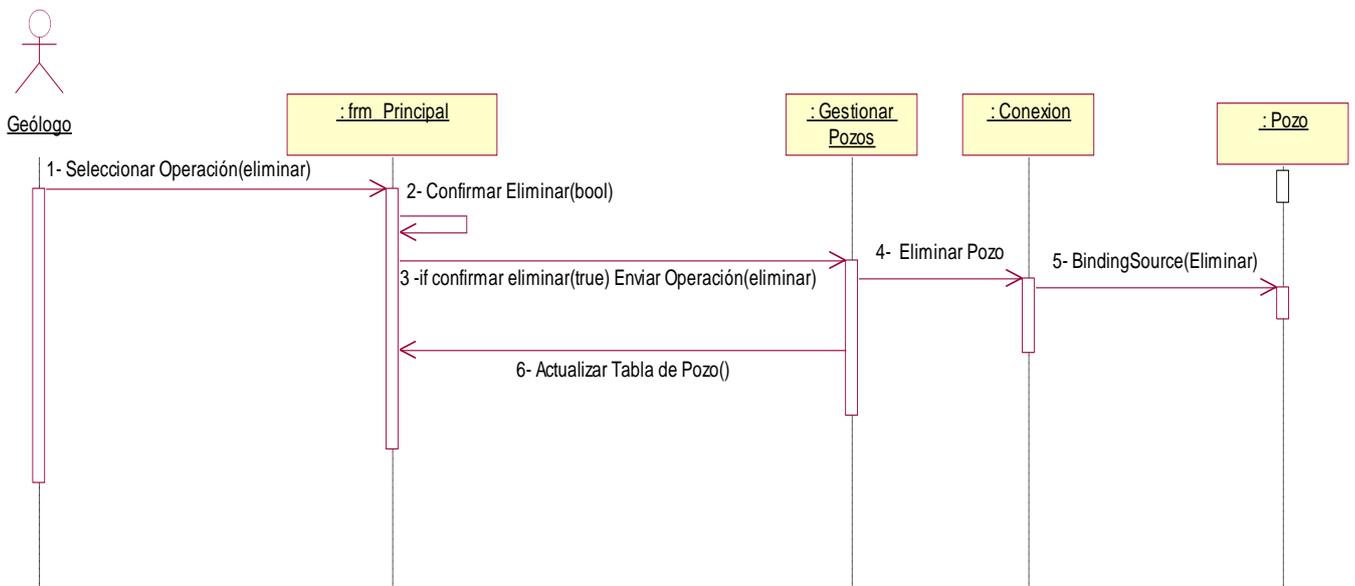


Fig. 5.2 Diagrama de Secuencia Eliminar Pozo

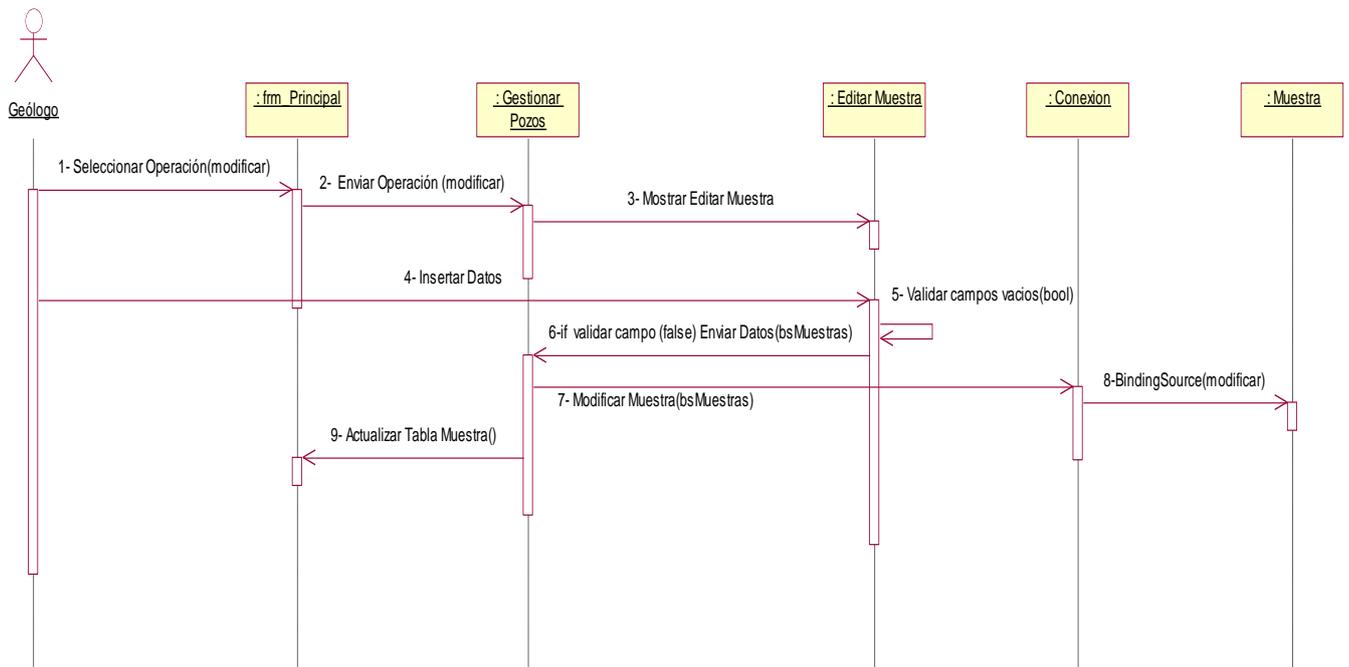


Fig. 5.3 Diagrama de Secuencia Modificar Muestra

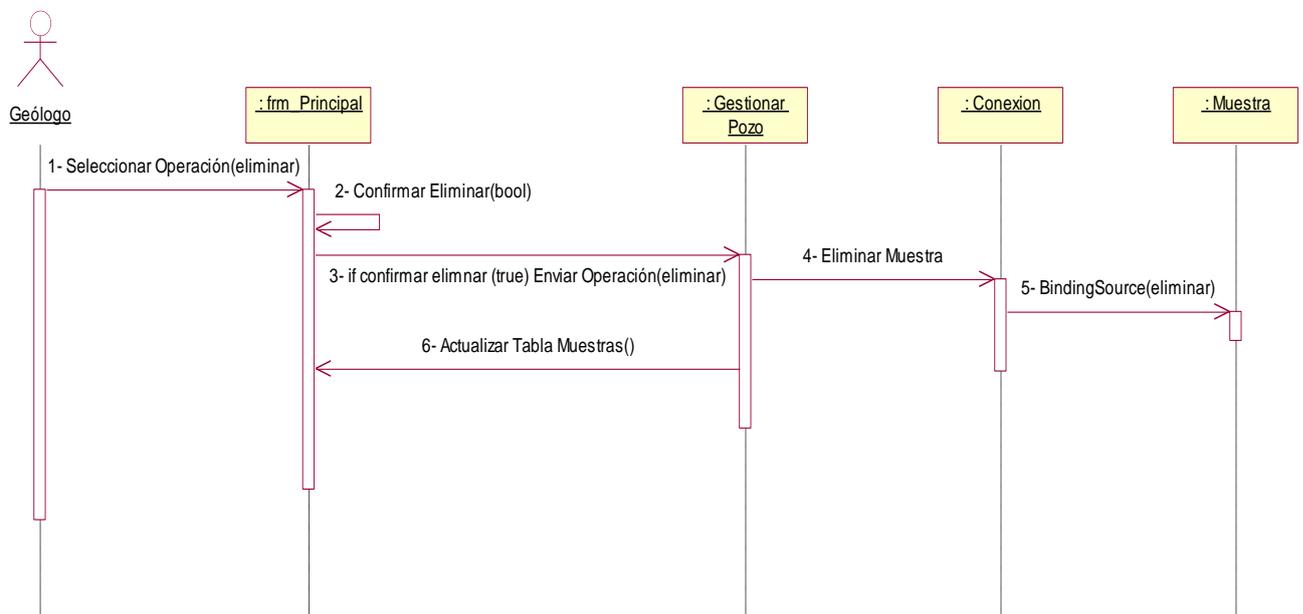


Fig. 5.4 Diagrama de Secuencia Eliminar Muestra

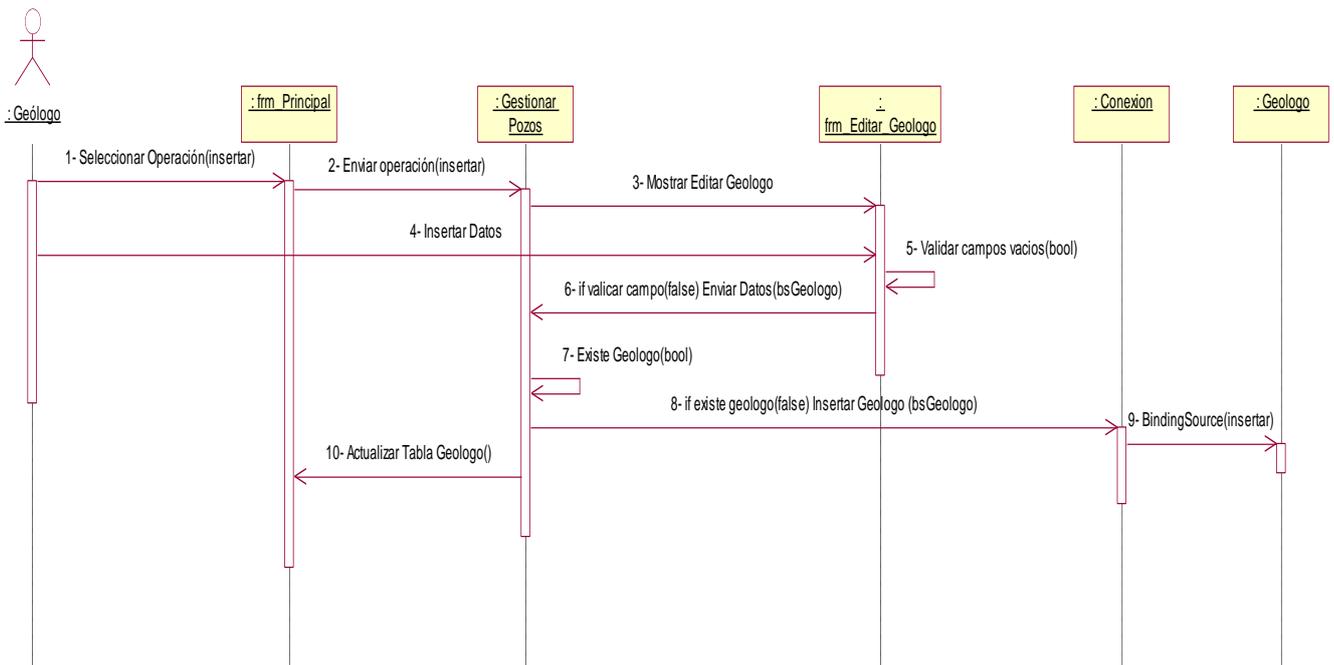


Fig. 5.5 Diagrama de Secuencia Insertar Geólogo

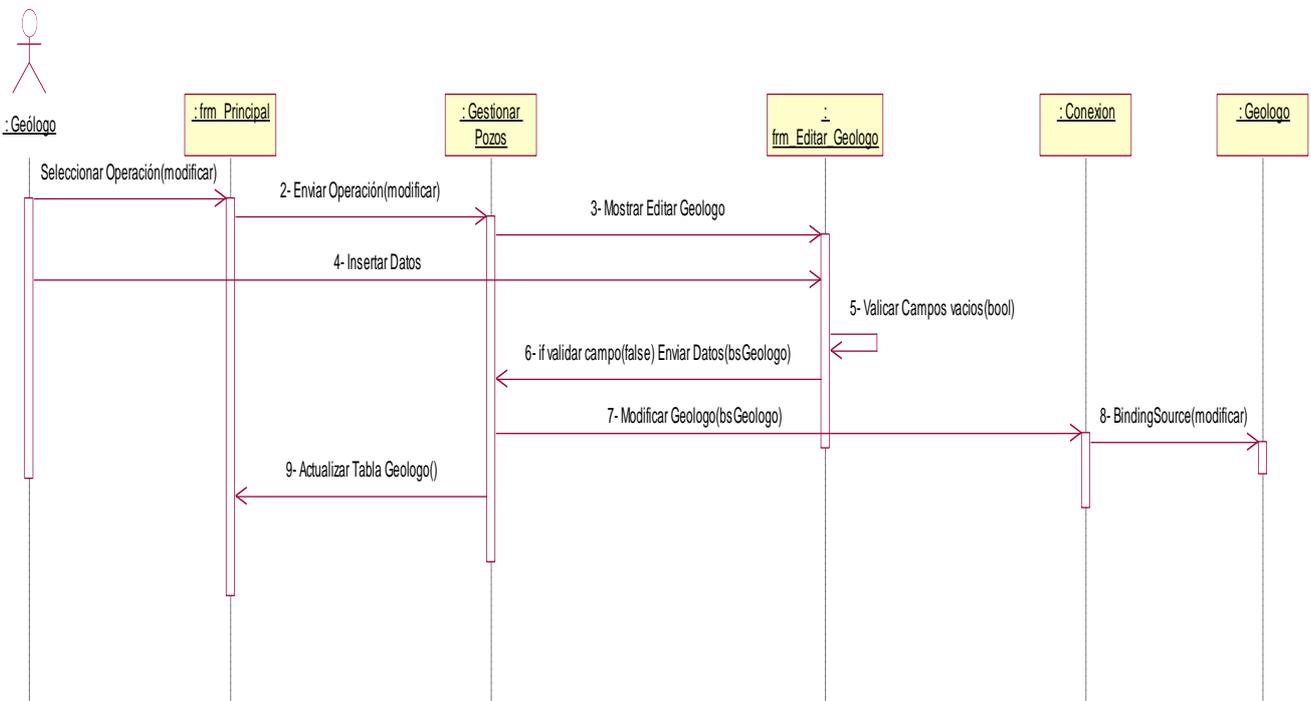


Fig. 5.6 Diagrama de Secuencia Modificar Geólogo

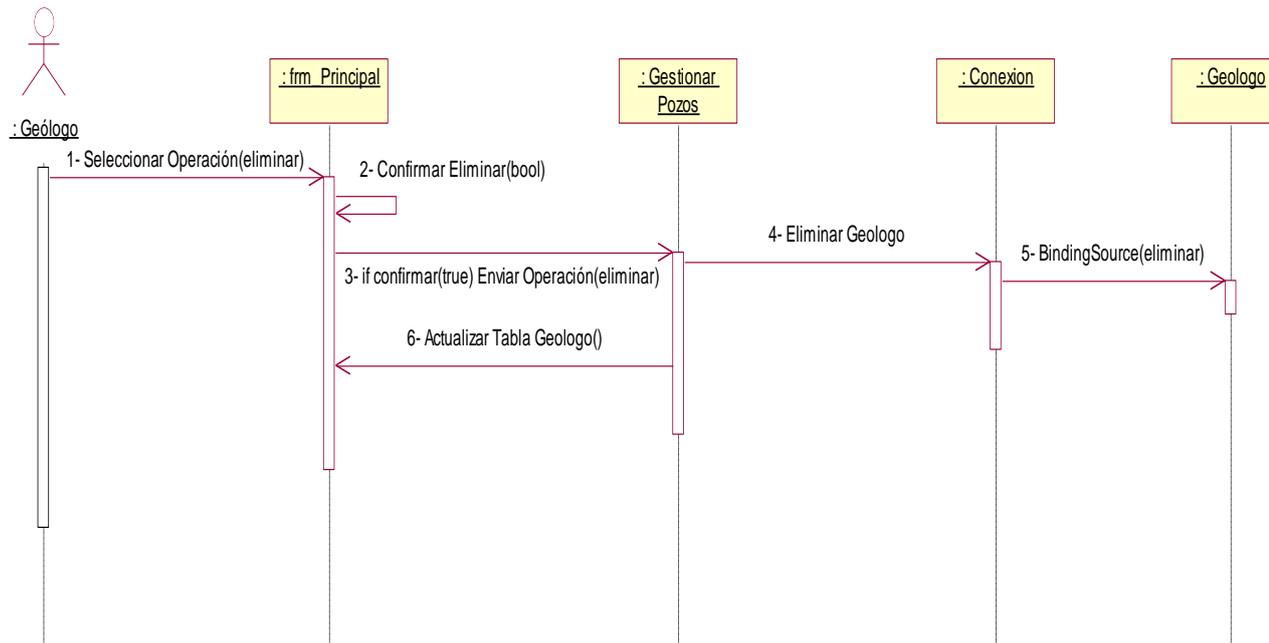


Fig. 5.7 Diagrama de Secuencia Eliminar Geólogo

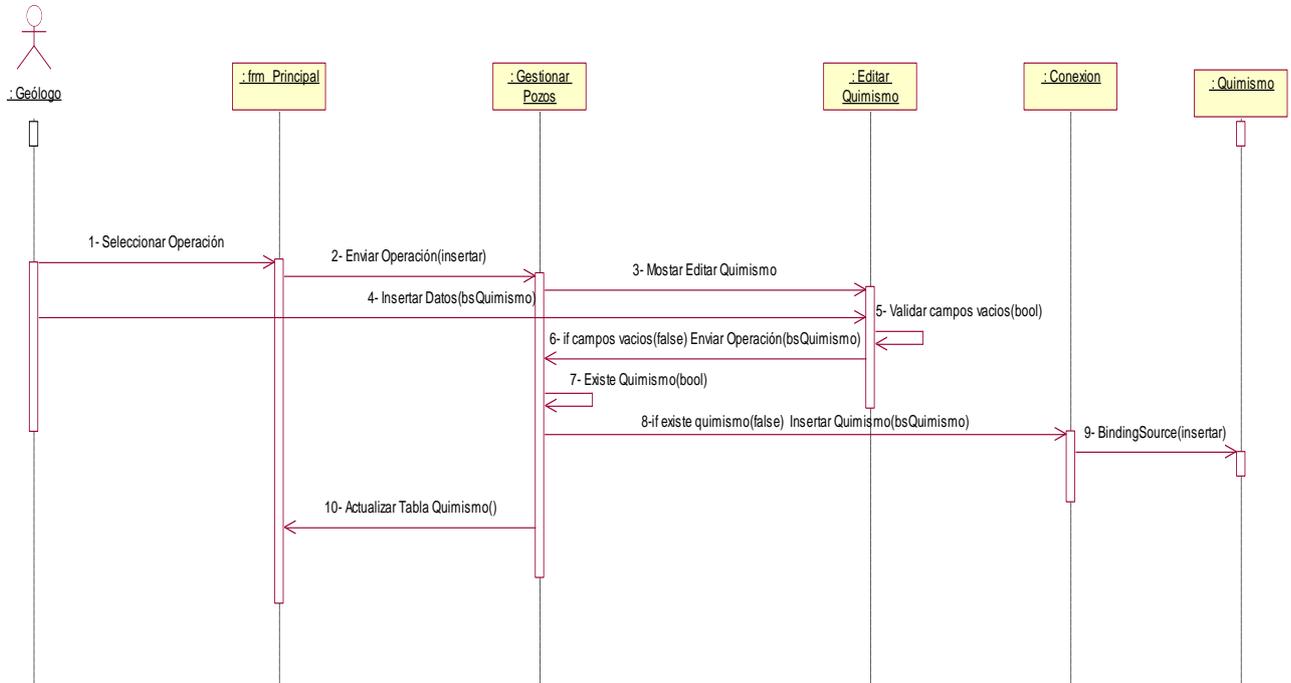


Fig. 5.8 Diagrama de Secuencia Insertar Quimismo

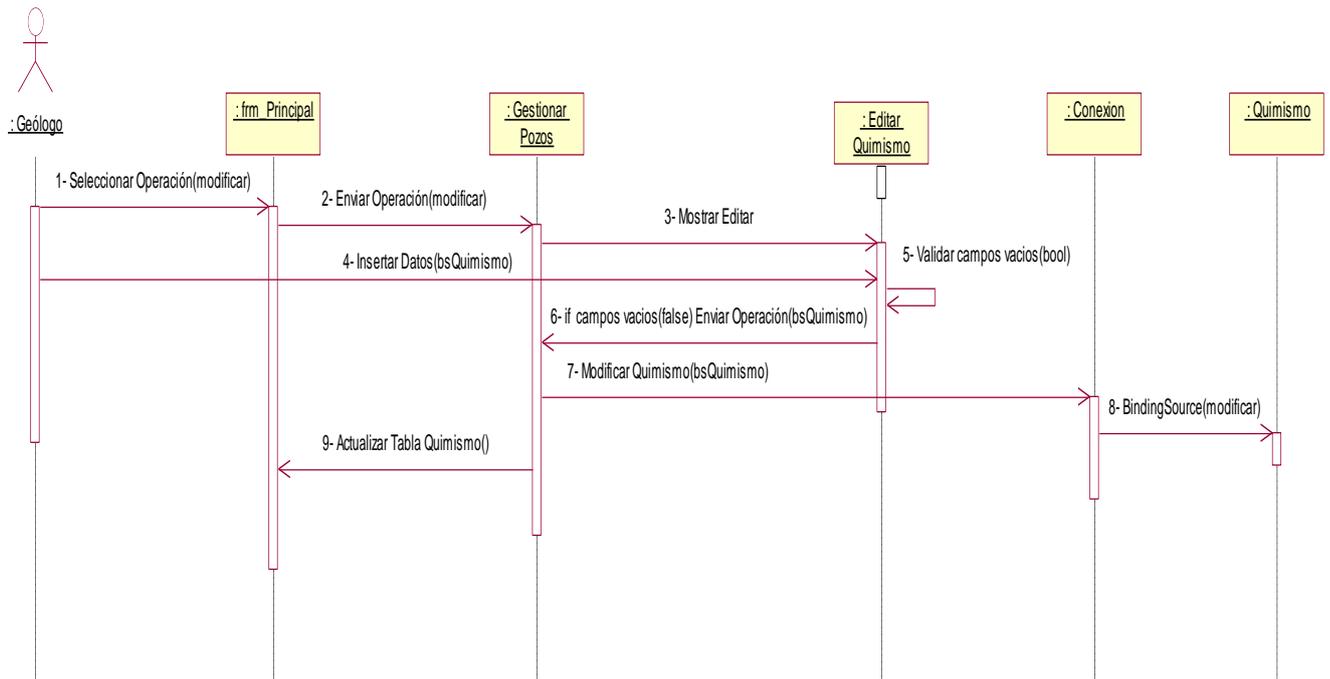


Fig. 5.9 Diagrama de Secuencia Modificar Quimismo

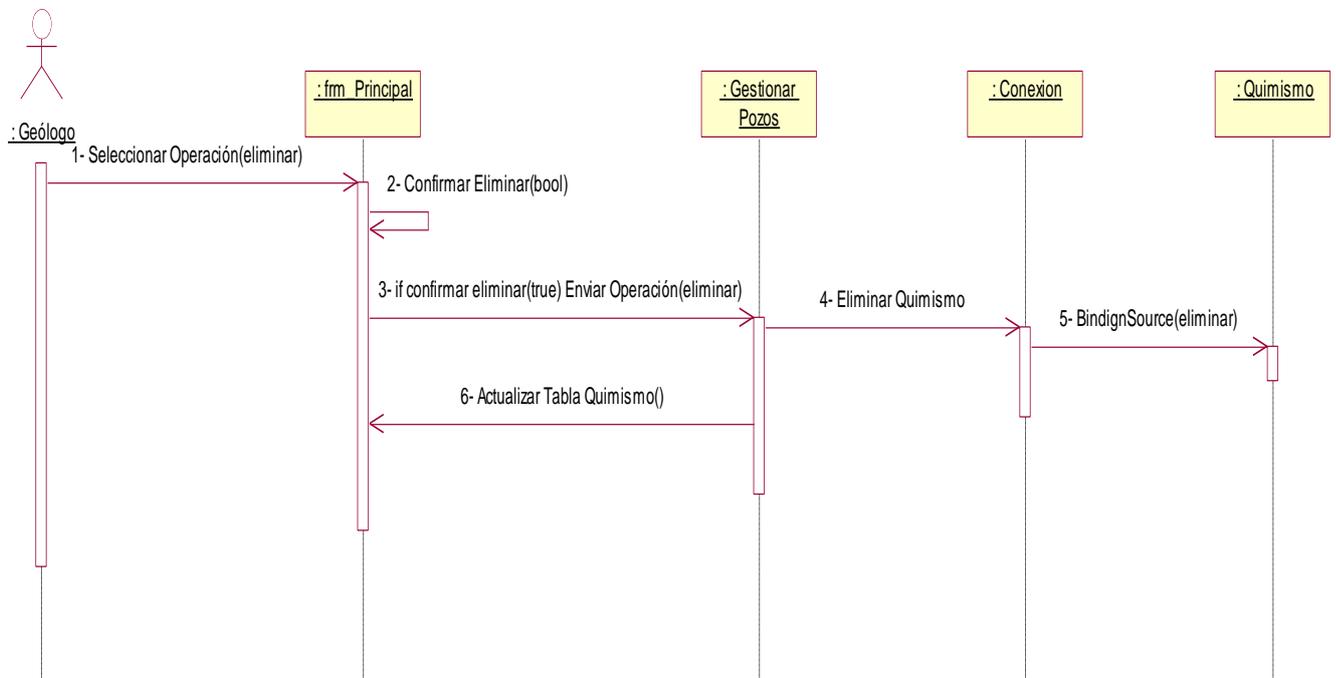


Fig. 6.1 Diagrama de Secuencia Eliminar Quimismo

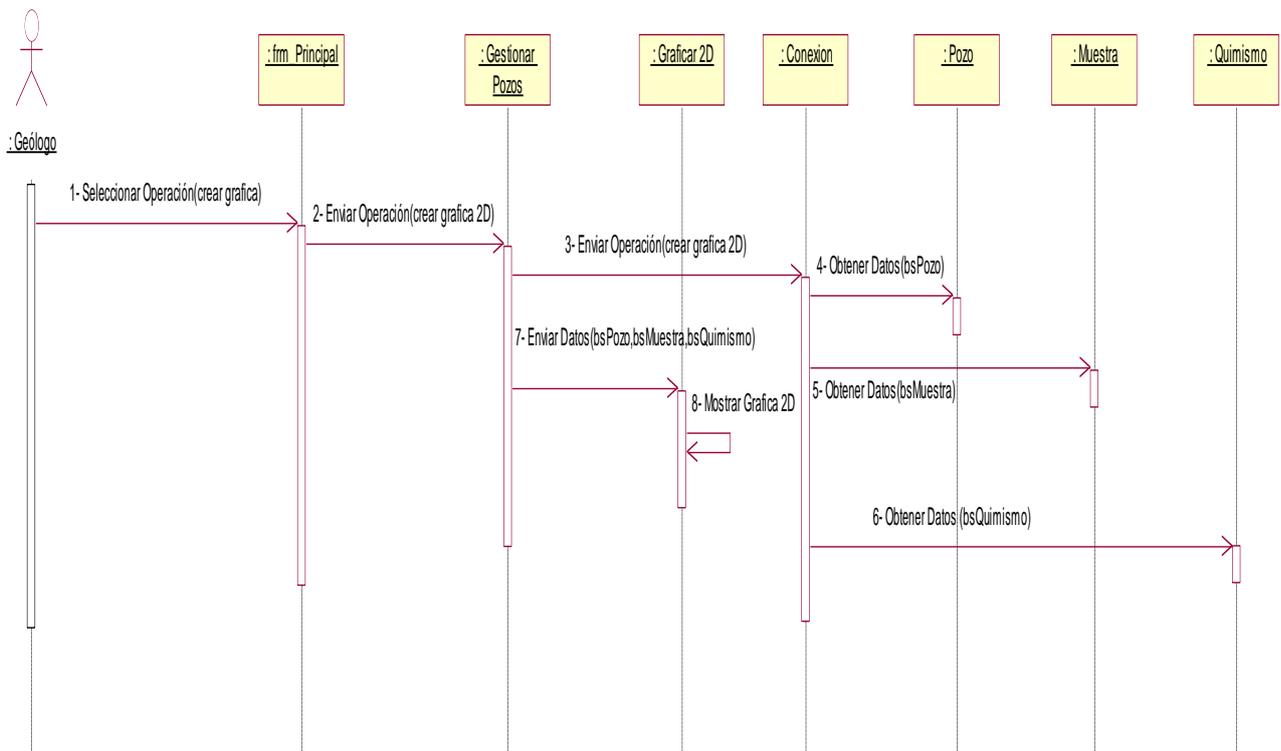


Fig. 6.2 Diagrama de Secuencia Crear Gráfica 2D

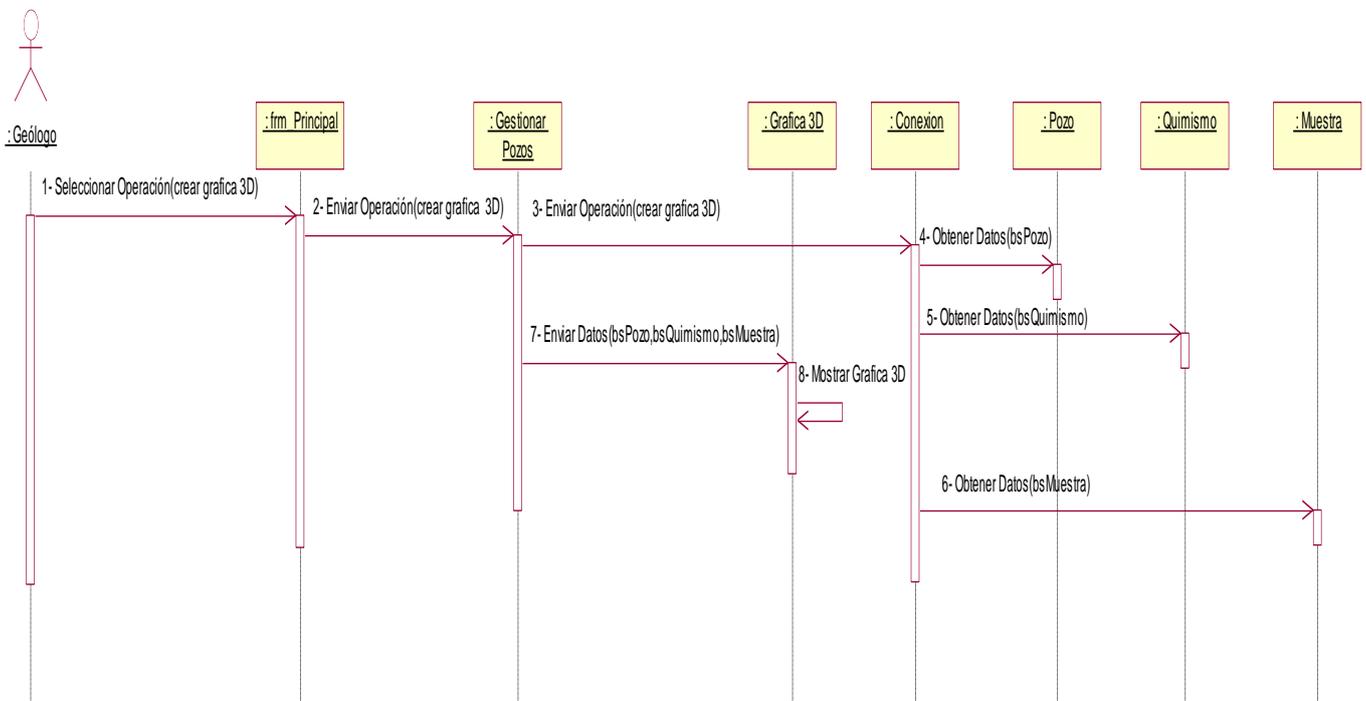


Fig. 6.3 Diagrama de Secuencia Crear Gráfica 3D

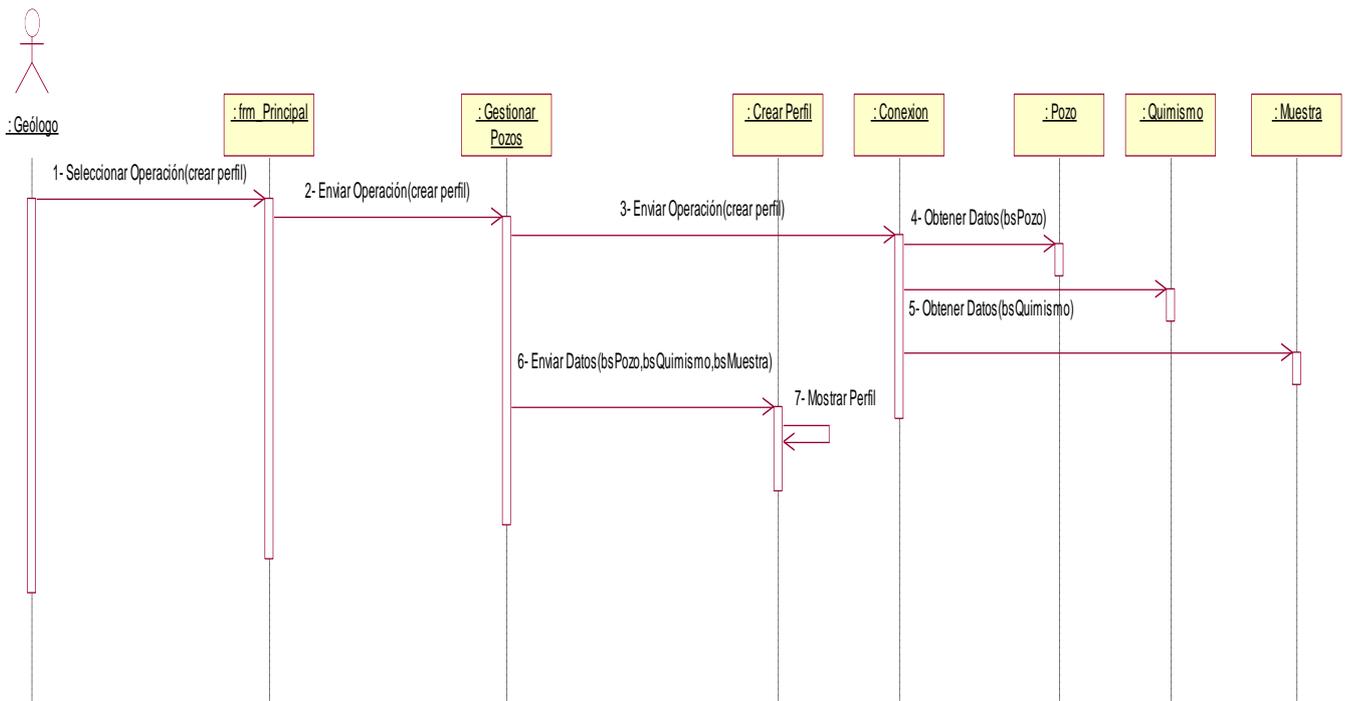


Fig. 6.4 Diagrama de Secuencia Crear Perfil Litológico

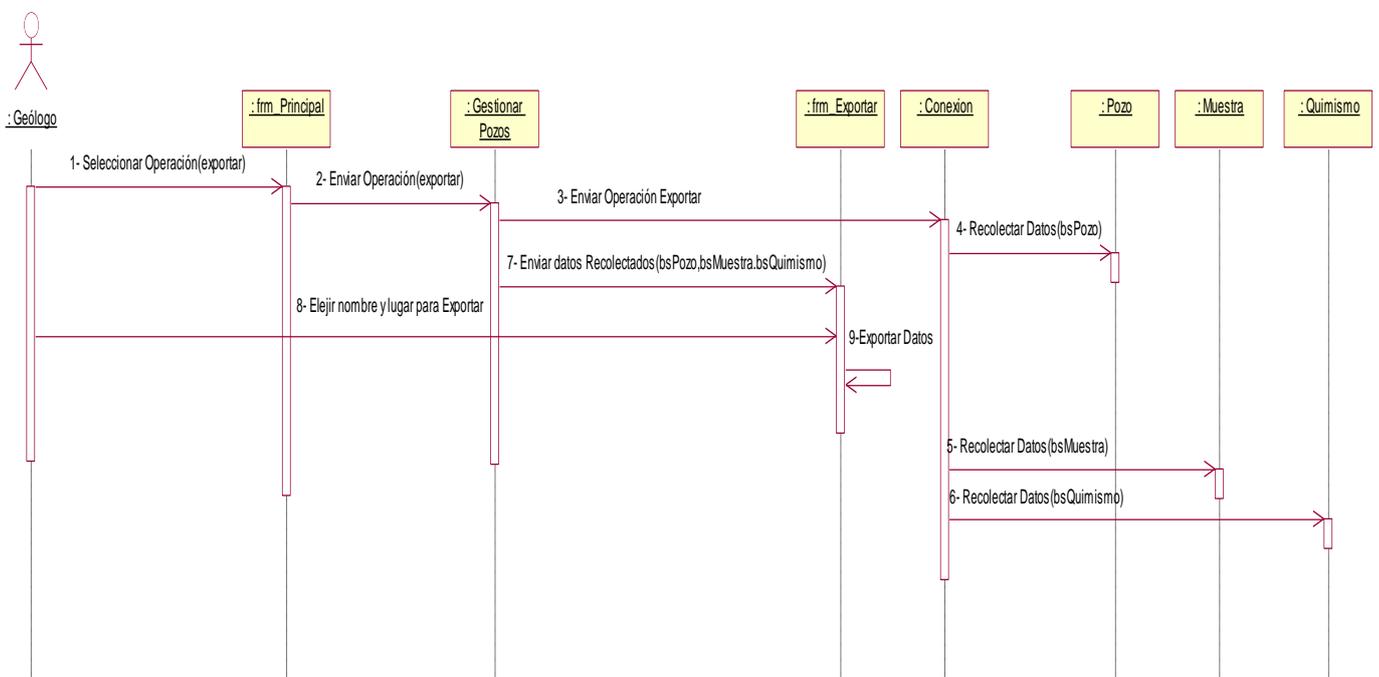


Fig. 6.5 Diagrama de Secuencia Exportar

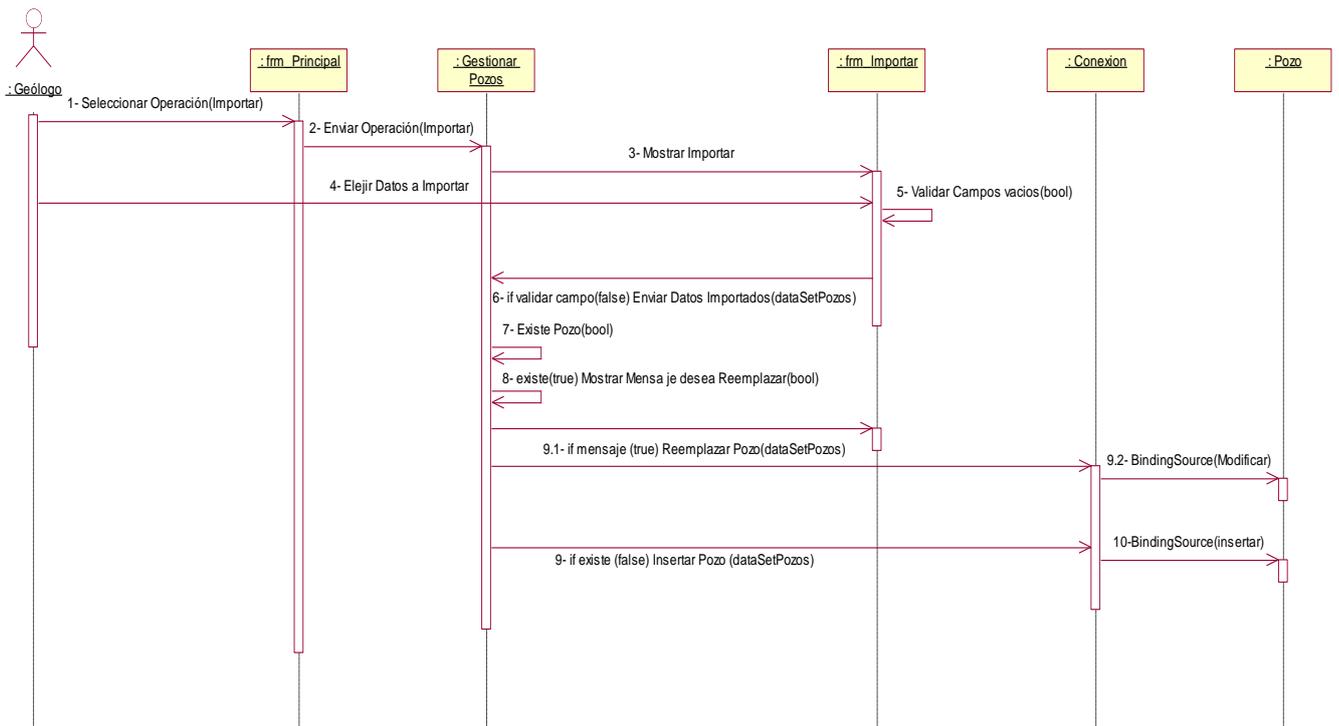


Fig. 6.6 Diagrama de Secuencia Importar Pozo

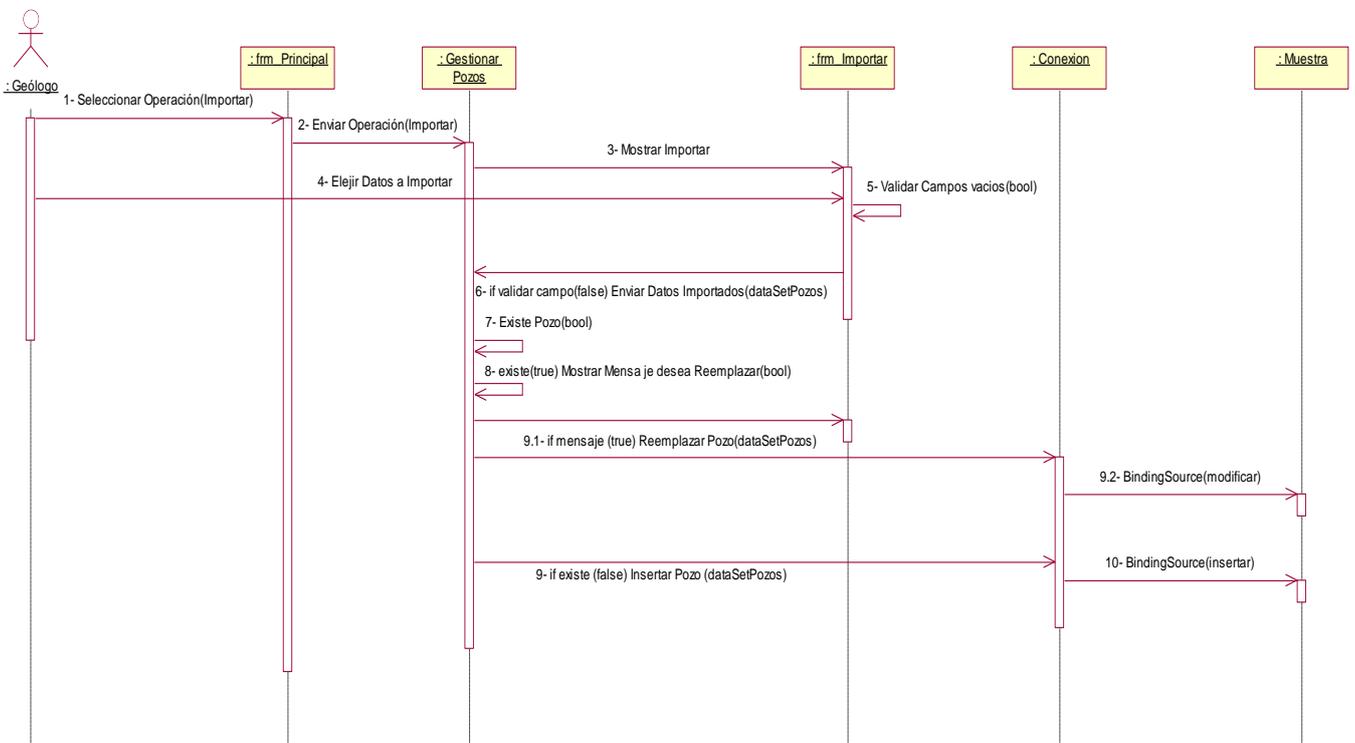


Fig. 6.7 Diagrama de Secuencia Importar Muestra

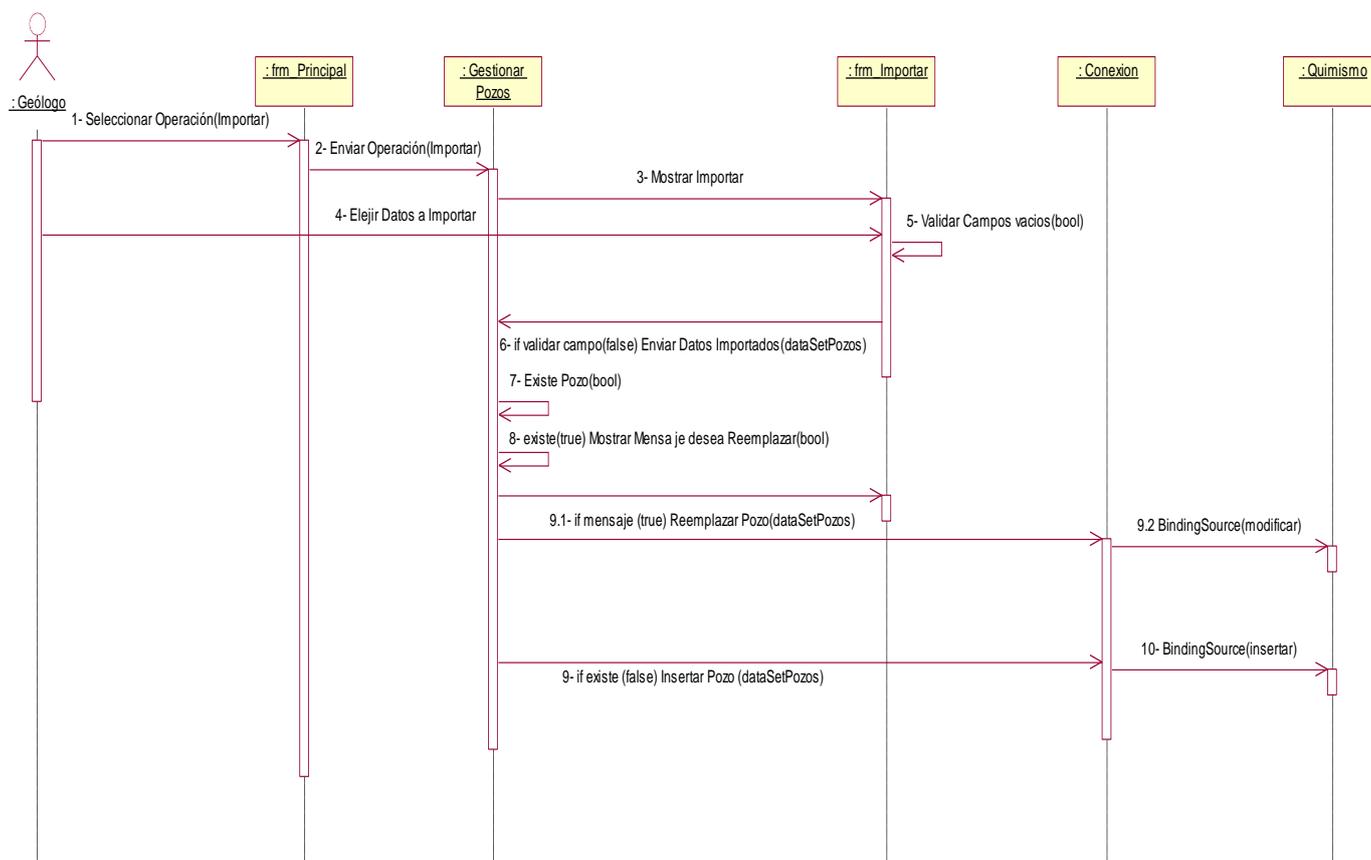


Fig. 6.8 Diagrama de Secuencia Importar Quimismo