

UNIVERSIDAD DE LAS CIENCIAS

INFORMÁTICAS

FACULTAD 6



TÍTULO:

Herramienta para la representación de los perfiles de los pozos de perforación minera

AUTORES:

Yoan Mandina Verdecia

Carlos Alberto Otero Alguacil

TUTORES:

MSc. Mario Pupo Meriño

Ing. Manuel Alejandro Pérez

La Habana, junio de 2014

“Año 56 de la Revolución”



“Si la belleza seduce al corazón, el triunfo es el que convence a la razón.”

José Raúl Capablanca y Graupera

DECLARACIÓN DE AUTORÍA

Declaramos por este medio que nosotros, Yoan Mandina Verdecia y Carlos Alberto Otero Alguacil con carné de identidad 90042827945 y 89012343728 respectivamente, somos los autores de este trabajo y que autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firma la presente declaración jurada de autoría en La Habana a los 16 días del mes de junio del año 2014.

Yoan Mandina Verdecia

Carlos Alberto Otero Alguacil

Autores

MSc. Mario Pupo Meriño

Ing. Manuel Alejandro Pérez

Tutores

AUTORES

Nombre: Yoan

Apellidos: Mandina Verdecia

Teléfono: 53060643

Nombre: Carlos Alberto

Apellidos: Otero Alguacil

Teléfono: 54260529

TUTORES

Nombre: Manuel Alejandro

Apellidos: Pérez Rosabal

Teléfono: 58180327

Correo electrónico: marosabal@uci.cu

Título de la especialidad: Ingeniería en Ciencias Informáticas

Año de graduación: 2012

Nombre: Mario

Apellidos: Pupo Meriño

Correo electrónico: mpupom@uci.cu

Título de la especialidad: Licenciado en Bioquímica, Máster en Matemática Aplicada

Categoría docente: Asistente

Categoría científica: Investigador Agregado

Año de graduación: 2005

AGRADECIMIENTOS

A mis padres Delia Verdecia y Paulino Mandina, por ser padres maravillosos que me lo han dado todo y siempre han confiado en mí.

A mi hermano por estar siempre pendiente y ser mi principal fuente de inspiración para lograr convertirme en ingeniero.

A mis tutores Manuel Rosabal y Mario Pupo, especialmente a Manuel por todo su apoyo, confianza y ayuda que hicieron posible la culminación de este trabajo.

A mi tía Lidia, mi tío Osvaldo, mi cuñada Yanet, a mis primos David, Yenisel, Dayana y a toda mi familia en general.

A mi novia María del Carmen por estar conmigo en todo momento, por su dedicación y comprensión.

Al tribunal de tesis y especialmente a la profesora Milenis por todo su apoyo incondicional del cual estaré siempre agradecido.

A los profesores del proyecto que de una forma u otra aportaron su granito de arena.

A todos mis amigos y amigas que me acompañaron en los buenos y malos momentos, y a todos aquellos que aportaron ideas y consejos para llevar a cabo esta obra.

Muchas Gracias Yoan.

A mis compañeros y amigos que siempre estuvieron conmigo ofreciendo su apoyo en todos estos años.

A mi suegro María Isabel y Miguel Ángel por acogerme como un hijo.

A Celso por estar presente en los momentos más difíciles.

A toda mi familia en general por estar siempre pendiente a todas mis necesidades.

Muchas Gracias Carlos Alberto.

DEDICATORIA

YOAN

A mi sobrinito Paulo Daríel, que espero que en un futuro se convierta en un buen profesional y a mi tía Margot que aunque ya no se encuentre físicamente, hubiese estado orgullosa.

CARLOS ALBERTO

A mis padres Beatriz y Benito por el apoyo incondicional en todos estos años y a mi novia Hanny por toda su comprensión y ayuda.

Durante el proceso de explotación minera una de las obras que realizan los especialistas mineros es la perforación de pozos, donde se adquieren datos acerca de las formaciones que se van atravesando. Estos datos son conocidos como información geológica minera, caracterizada por poseer volúmenes considerables de datos y brindar poca información en su forma natural, lo que dificulta el estudio y análisis de los yacimientos minerales por parte de los especialistas mineros. En el desarrollo de software para la minería es fundamental y de gran importancia el modelado geológico que no es más que la creación de un modelo numérico que represente en lo posible la realidad de un subsuelo. Este permite crear a partir de la información geológica minera, los perfiles de los pozos de perforación minera, los cuales incluyen las capas litológicas y la concentración de los diferentes minerales que posee el pozo en su trayectoria. Con la presente investigación se desarrolló una aplicación informática que permite la representación de los perfiles de los pozos de perforación minera para el estudio y análisis de los yacimientos minerales. Esta herramienta fue construida utilizando la biblioteca *QCustomPlot*, el *framework* de desarrollo *Qt*, el lenguaje de programación *C++*, el sistema gestor de bases de datos *PostgreSQL* y la metodología de desarrollo de software *XP*. De la explotación futura de esta herramienta se derivan ventajas importantes como son: clasificar y evaluar la potencialidad del yacimiento mineral, identificar formaciones problemáticas y mitigar posibles problemas futuros.

Palabras clave:

Información geológica minera, perfil de pozo, modelado geológico, yacimiento mineral.

During the mining process one of the tasks performed by mining specialists is drilling wells, where data is acquired about the formations that are going through. These data are known as mining geological information, and they are characterized by having large volumes of data and provides little information in its natural form, which makes difficult the study and analysis of mineral deposits by mining specialists. In mining for software development the geological modeling is fundamental and very important that is the creation of a numerical model to represent the possible reality of subsoil. This research allows creating from mining geological information, profiles of mining drilling wells, which includes lithological layers and concentration of different minerals that wells have in its trajectory. With this research, it is intended to develop computer application that allows the representation of profiles of mining drilling wells for the study and analysis of mineral deposits. This tool was built using QCustomPlot library, the development framework Qt, the programming language C++, the database management system PostgreSQL and the software development methodology XP. From the further use of this tool some important advantages will be obtained such as: classify and evaluate the potential of the mineral deposit, identify complicated formations and mitigate potential future problems.

Keywords:

Geological modeling, mineral deposit, mining geological information, profile well.

ÍNDICE DE CONTENIDOS

| | |
|--|----|
| Introducción..... | 1 |
| CAPÍTULO I: Fundamentos de la investigación, metodología y tecnologías empleadas para la representación de los perfiles de los pozos de perforación minera | 6 |
| 1.1 Proceso de Modelado Geológico..... | 6 |
| 1.1.1. Representación de los perfiles de los pozos de perforación minera..... | 7 |
| 1.2 Análisis de otras soluciones existentes | 9 |
| 1.3 Lenguajes de Programación | 12 |
| 1.3.1. C++ como lenguaje de programación | 13 |
| 1.4 Framework de desarrollo Qt | 13 |
| 1.5 Biblioteca QCustomPlot | 14 |
| 1.6 Sistemas Gestores de Bases de Datos..... | 16 |
| 1.6.1. PostgreSQL como Sistema Gestor de Bases de Datos | 17 |
| 1.7 Metodologías de desarrollo de software..... | 18 |
| 1.7.1. Programación Extrema (XP) como metodología de desarrollo de software | 18 |
| 1.8 Lenguaje de Modelado..... | 20 |
| 1.8.1. Lenguaje Unificado de Modelado (UML)..... | 20 |
| 1.9 Herramienta CASE..... | 21 |
| 1.9.1. Visual Paradigm como herramienta CASE | 22 |
| 1.10 Conclusiones..... | 23 |
| CAPÍTULO II: Características de la herramienta para la representación de los perfiles de los pozos de perforación minera | 24 |
| 2.1. Modelo de dominio | 24 |
| 2.1.1. Descripción del modelo de dominio | 25 |
| 2.2. Descripción de la solución | 26 |
| 2.3. Fase de Diseño | 26 |
| 2.3.1. Historias de usuarios | 26 |
| 2.3.2. Lista de reserva de producto | 31 |
| 2.3.3. Plan de iteraciones | 32 |
| 2.3.4. Tarjetas CRC..... | 33 |
| 2.4. Arquitectura base de la aplicación..... | 35 |
| 2.4.1. Patrones de arquitectura | 35 |

ÍNDICE DE CONTENIDOS

| | |
|---|----|
| 2.4.2. Patrones de diseño..... | 37 |
| 2.5. Conclusiones..... | 41 |
| CAPÍTULO III: Implementación y evaluación de la herramienta para la representación de los perfiles de los pozos de perforación minera | 42 |
| 3.1. Implementación de la aplicación..... | 42 |
| 3.1.1. Tareas de ingeniería..... | 42 |
| 3.1.2. Estándares de codificación..... | 45 |
| 3.1.3. Interfaces de la aplicación..... | 46 |
| 3.2. Proceso de pruebas de la metodología XP | 49 |
| 3.2.1. Tipos de pruebas | 49 |
| 3.2.2. Técnicas de pruebas | 50 |
| 3.3. Descripción del proceso de pruebas | 51 |
| 3.4. Análisis de los resultados del proceso de prueba | 55 |
| 3.5. Conclusiones..... | 56 |
| Conclusiones Generales..... | 57 |
| Recomendaciones | 58 |
| Referencias Bibliográficas | 59 |
| Bibliografía Consultada..... | 61 |
| Anexos | 64 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Historia de Usuario #1. ----- | 27 |
| Tabla 2. Historia de Usuario #2. ----- | 28 |
| Tabla 3. Historia de Usuario #3. ----- | 30 |
| Tabla 4. Fragmento de LRP. ----- | 31 |
| Tabla 5. Plan de iteraciones. ----- | 33 |
| Tabla 6. Tarjeta CRC# 1.----- | 34 |
| Tabla 7. Tarjeta CRC# 2.----- | 34 |
| Tabla 8. Tarea de ingeniería #1.----- | 43 |
| Tabla 9. Tarea de ingeniería #2.----- | 43 |
| Tabla 10. Tarea de ingeniería #3. ----- | 44 |
| Tabla 11. Tarea de ingeniería #4. ----- | 44 |
| Tabla 12. Caso de prueba CP1_HU1.----- | 52 |
| Tabla 13. Caso de prueba CP1_HU2.----- | 53 |
| Tabla 14. Caso de prueba CP1_HU3.----- | 54 |
| Tabla 15. Resultados de los casos de pruebas aplicados. ----- | 55 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Fig. 1.-Herramienta <i>LogPlot 7</i> . ----- | 10 |
| Fig. 2.-Herramienta <i>DownHole Explorer</i> . ----- | 12 |
| Fig. 3.-Ejemplos de la biblioteca <i>QCustomPlot</i> .----- | 15 |
| Fig. 4.-Modelo de Dominio. ----- | 25 |
| Fig. 5. -Diseño de la arquitectura. ----- | 37 |
| Fig. 6.-Patrón Creador. ----- | 38 |
| Fig. 7.-Patrón Controlador.----- | 38 |
| Fig. 8.-Patrón instancia única.----- | 40 |
| Fig. 9.-Patrón D-pointer. ----- | 40 |
| Fig. 10.-Estructura de una clase. ----- | 45 |
| Fig. 11.-Interfaz para conectar aplicación con la base de datos.----- | 47 |
| Fig. 12.-Interfaz que muestra cómo se importan los datos desde un archivo. ----- | 47 |
| Fig. 13.-Interfaz que muestra el perfil litológico de un pozo determinado. ----- | 48 |
| Fig. 14.-Interfaz que muestra la estructura litológica del terreno. ----- | 48 |
| Fig. 15.-Interfaz que muestra la localización de los pozos en el terreno. ----- | 49 |
| Fig. 16.-Tipos de pruebas de caja blanca. ----- | 51 |
| Fig. 17.-Tipos de pruebas de caja negra.----- | 51 |

Introducción

Hoy día, es indiscutible que la exploración y explotación de determinado mineral¹ en la naturaleza es de considerable importancia para el porvenir económico de un país. Muchos de los objetos y herramientas que se utilizan en la vida cotidiana y que son indispensables para la humanidad provienen de ciertos minerales, algunos de los más conocidos y utilizados son oro, plata, cobre, hierro, zinc, plomo, níquel, entre muchos otros. Es por ello, que el desarrollo de la industria geólogo-minera es uno de los sectores económicos de considerable valor para algunas naciones e indispensable para otras. Cuba no se encuentra alejada de esta realidad puesto que cuenta con una gran variedad de recursos minerales, de ahí que se decida impulsar el sector minero en el país con el objetivo de desarrollar la industria nacional.

El modelado es el cimiento de la estimación de recursos, es un trabajo multidisciplinario que requiere de la interacción de varias ciencias y un conjunto de tareas como son el procesamiento de datos, control de la calidad, análisis de la incertidumbre, estudio de la sensibilidad numérica, entre muchas otras. El principal objetivo del modelado geológico es la creación de un modelo numérico que represente en lo posible la realidad de un subsuelo. El modelado geológico es fundamental y de gran importancia ya que facilita la toma de decisiones a los especialistas (Tomás y otros, 2004).

En el transcurso de los años Cuba ha contado con un conjunto de aplicaciones de procedencia nacional para facilitar el trabajo de las empresas mineras, entre estas se pueden mencionar MICRONIQ, Software Integral Minero (SIM) y TIERRA que en su momento fueron de gran ayuda para las empresas cubanas. Estas pequeñas aplicaciones no satisfacían completamente a las empresas cubanas, por lo que algunas de ellas se vieron obligadas a adquirir licencias de productos internacionales que contaban con todas las características para asegurar el éxito en el negocio minero (Tercera convención cubana de las ciencias de la tierra, 2009). Actualmente a las empresas mineras se les hace difícil adquirir estas soluciones, mantenerlas actualizadas y sobre todo distribuir las, por ser herramientas privativas y tener costos elevados, de ahí que se encuentran realizando grandes esfuerzos por automatizar todos sus servicios, y

¹ Sustancia inorgánica que se halla en el suelo o en el subsuelo, y principalmente aquella cuya explotación ofrece interés económico.

como estrategia comenzaron a realizar colaboraciones con empresas e instituciones productoras de software en Cuba.

Durante el proceso de explotación minera, una de las obras que realizan los especialistas mineros es la perforación de pozos donde se adquiere información acerca de las características de las formaciones que se van atravesando. La estratigrafía es la rama de la geología que se encarga del análisis y estudio de las rocas que van surgiendo en estas perforaciones. Una de las áreas que estudia la estratigrafía es la litoestratigrafía, la cual caracteriza la litología² de las diferentes sucesiones estratigráficas (Dabrio y otros, 2003). Durante la perforación de un pozo se obtiene la información geológica minera correspondiente a este, que no es más que una grabación contra profundidad de las capas litológicas, los minerales y sus características a partir de técnicas y herramientas de medición.

Los especialistas mineros están interesados en analizar de la información geológica minera, la litología y la concentración de todos los minerales que van surgiendo en el proceso de perforación de los pozos. Una columna estratigráfica (columna litológica, perfil litológico) es una representación utilizada en geología específicamente en la rama de la estratigrafía para representar y describir la ubicación vertical de las rocas en un área determinada, que no es más que la estratigrafía generalizada de un sector.

Estos datos de muestras de rocas recuperadas de una formación de interés juegan un papel vital en los programas de exploración, operaciones de completado, reacondicionamiento de pozos, análisis de datos, estudios comparativos, análisis de riesgos y otras. Estos datos también permiten a los perforadores identificar formaciones problemáticas y mitigar posibles problemas futuros en otras perforaciones en lugares equivalentes, además ofrece la posibilidad de clasificar y evaluar la potencialidad del yacimiento. Estos registros también son necesarios para estudiar un conjunto de propiedades de las rocas que se van perforando, algunas de las más importantes son: definición de los límites de las capas de las rocas, detección de las rocas mineralizadas y su espesor, detección de fugas de acuíferos³ de agua dulce y salada, confirmación de si el acuífero es roca porosa o roca fracturada, entre otras características de elevada importancia para el análisis de los especialistas mineros (Dabrio y otros, 2003).

² Parte de la geología que trata sobre las rocas.

³ Capa subterránea de roca permeable que almacena, filtra y libera agua.

Estas muestras obtenidas componen grandes volúmenes de datos debido a que para la eficiente explotación de un yacimiento mineral es necesario perforar un conjunto considerable de pozos. Estos datos obtenidos de las muestras realizadas a los pozos brindan poca información en su forma natural, lo que dificulta un estudio y análisis más profundo y real de los yacimientos minerales por parte de los especialistas mineros. Para establecer comparaciones y facilitar el trabajo de los geólogos es más factible proporcionar estos datos obtenidos de los pozos de perforación en un formato gráfico, esto constituye una aproximación a escala de la realidad, realizándose anotaciones donde se ubican los diferentes tipos de litologías que son atravesadas y la concentración de cada uno de los minerales que posee. Los datos proporcionados en un formato gráfico permiten al geólogo captar información de forma visual que de otra manera fuera muy difícil de obtener.

Es por ello que a partir de la situación problemática anteriormente descrita, se determinó el siguiente **problema a resolver**: Los grandes volúmenes de datos y la poca información que brindan en su forma natural, dificultan el estudio y análisis de los yacimientos minerales.

La investigación tiene como **objeto de estudio**: El proceso de modelado geológico enmarcado en el **campo de acción**: la representación de los perfiles de los pozos de perforación minera.

Para dar solución al problema antes expuesto se define como **objetivo general**: Implementar una herramienta informática que permita la representación de los perfiles de los pozos de perforación minera para el estudio y análisis de los yacimientos minerales.

Se plantean las siguientes **preguntas científicas**:

1. ¿Cuáles son las características de la información geológica minera y que elementos componen los perfiles de los pozos de perforación minera?
2. ¿Cuáles son las soluciones existentes para representar los perfiles de los pozos de perforación minera?
3. ¿Cuáles son las características que debería tener una herramienta de representación de los perfiles de los pozos de perforación minera para facilitar el estudio y análisis de los yacimientos minerales?
4. ¿Cómo implementar una herramienta que represente los perfiles de los pozos de perforación minera para facilitar el estudio y análisis de los yacimientos minerales?

5. ¿Cómo determinar si la herramienta para la representación de los perfiles de los pozos de perforación minera cumple correctamente con las características que se esperan de este?

Para el cumplimiento del objetivo general se trazaron las siguientes **tareas de la investigación**:

1. Caracterización de la información geológica minera, los algoritmos y técnicas de representación de los perfiles de los pozos de perforación minera.
2. Caracterización de las principales tecnologías y herramientas que se utilizarán para la construcción de la solución propuesta.
3. Caracterización de las soluciones existentes e identificación de las funcionalidades básicas que serán implementadas en la herramienta propuesta.
4. Elaboración de la documentación técnica asociada a la metodología de desarrollo utilizada.
5. Implementación de la herramienta propuesta.
6. Realización de las pruebas necesarias a la solución propuesta para validar su correcto funcionamiento.

Los métodos científicos de la investigación utilizados fueron:

Métodos teóricos:

Analítico-Sintético: Se profundizó en la teoría y se realizó un análisis, obteniéndose una síntesis detallada de la representación de los perfiles de los pozos de perforación minera en el proceso de modelado geológico. Se analizaron las tendencias generales para resolver este tipo de problemas y se obtuvo un esquema propio de solución.

Análisis Histórico-Lógico: Este método permitió estudiar la trayectoria, evolución y desarrollo de la representación de los perfiles de los pozos de perforación minera, así como la existencia de soluciones de este tipo implementados a nivel nacional e internacionalmente y de este modo conocer su funcionamiento. A través de este método se registraron y se descubrieron las características más importantes en el proceso de representación de los perfiles de los pozos de perforación minera.

Modelación: Se utilizó en la modelación de diagramas dentro de la metodología de desarrollo de software Programación Extrema (*XP*) para darle solución al problema.

Una vez concluida la investigación se esperan obtener los siguientes **resultados**:

1. Una herramienta que represente los perfiles de los pozos de perforación minera a partir de su información geológica minera.
2. Documentación técnica asociada a la herramienta propuesta.

El presente documento está dividido en tres capítulos. A continuación se hace referencia al contenido que aborda cada uno de ellos.

Capítulo I: “Fundamentos de la investigación, metodología y tecnologías empleadas para la representación de los perfiles de los pozos de perforación minera.” En este capítulo se describe la representación de los perfiles de los pozos de perforación minera en el proceso de modelado geológico y se realiza un análisis de las soluciones existentes. Además se selecciona *XP* como metodología de desarrollo de software a utilizar y se seleccionan las herramientas, técnicas y tecnologías para darle solución al problema.

Capítulo II: “Características de la herramienta para la representación de los perfiles de los pozos de perforación minera.” En este capítulo se describe la solución propuesta y se realiza un estudio de los procesos del negocio, describiéndolos a través del modelo del dominio. Se identifican los requisitos funcionales y no funcionales a través de las historias de usuario. Además se presentan los artefactos obtenidos en el proceso ingenieril.

Capítulo III: “Implementación y evaluación de la herramienta para la representación de los perfiles de los pozos de perforación minera”. En este capítulo se procede a implementar el sistema confeccionando las tarjetas de clase, responsabilidad y colaboración, además de diseñar las tareas de ingeniería. También se abordan y analizan las estrategias de pruebas propias de *XP* y se seleccionan las que van a ser aplicadas en la herramienta para la representación de los perfiles de los pozos de perforación minera. Se describen las pruebas que se realizaron al software y se aborda un análisis de los resultados obtenidos en la aplicación de los casos de pruebas.

CAPÍTULO I: Fundamentos de la investigación, metodología y tecnologías empleadas para la representación de los perfiles de los pozos de perforación minera

Para lograr que los especialistas mineros realicen un correcto análisis de la información geológica minera es preciso realizar un modelado lo más exacto posible de los datos recolectados. De ahí la importancia de entender bien conceptos como modelado geológico, estratigrafía y perfil de pozo.

En el presente capítulo se realiza un análisis de las herramientas existentes para la representación de los perfiles de los pozos de perforación minera que pudieran ser útiles en el desarrollo de la herramienta que se tiene como propuesta de solución. Además se selecciona la metodología de desarrollo a utilizar en la investigación y se realiza un análisis de los lenguajes, sistemas, tecnologías y técnicas utilizadas para darle solución al problema.

1.1 Proceso de Modelado Geológico

Un modelo es una representación simplificada de la realidad, que se elabora para facilitar su comprensión y estudio, que permite ver de forma clara y sencilla las distintas variables y las relaciones que se establecen entre ellas. Un modelo geológico idealmente es una réplica virtual de lo que existe en la naturaleza, durante este proceso existe una interrelación entre la información, el conocimiento, la tecnología, la incertidumbre y los riesgos. El proceso de modelado geológico es multidisciplinario ya que requiere la interacción de varias disciplinas como la Geofísica, Geología⁴ Estructural, Sedimentología, Estratigrafía, Petrofísica, Estadística clásica, Geoestadística entre otras (Tomás y otros, 2004). Mediante la integración de datos geológicos, geoquímicos y geofísicos se construyen modelos numéricos que caracterizan y simulan en lo posible la realidad de un subsuelo. Los niveles de incertidumbre que se generan en los modelos son siempre un problema que el geólogo debe enfrentar, reducir y resolver para garantizar un alto grado de éxito en los proyectos de desarrollo. El proceso de modelado geológico se puede separar en tres ejes principales que recogen los elementos fundamentales (Tomás y otros, 2004):

⁴ Ciencia que analiza la forma interior y exterior del globo terrestre.

- 1.- Conceptualización geológica del depósito acorde con el conocimiento e información disponible.
- 2.- Definición de unidades de estimación trazables y reproducibles que maximicen la predictibilidad de los recursos estimados.
- 3.- Generación de modelos geológicos operativos bidimensionales y tridimensionales, espacialmente consistentes con la información disponible y conceptualización geológica.

En la investigación del subsuelo se obtienen muestras de suelo y roca de manera que se pueda determinar la clasificación o tipo de roca, su mineralogía y propiedades. El número, tipo y localización de las muestras debe definirse en función de generar la información necesaria para el diseño y la construcción del proyecto para el que se elabora el estudio. Cada muestra debe identificarse cuidadosamente con el número del respectivo pozo y la anotación de la profundidad en la cual fue tomada para su posterior análisis.

Los especialistas mineros tienen particular interés por analizar la litología y la concentración de todos los minerales que van surgiendo en el proceso de perforación de los pozos. Diversos tipos de minerales pueden ser obtenidos en los yacimientos, estos se pueden clasificar en metales, minerales industriales, minerales de construcción, gemas⁵ y combustibles. Estos minerales se identifican y estudian sus propiedades bajo tres subdivisiones principales, estas son: mineralogía química, mineralogía física y cristalografía⁶ (Mora, 2010).

1.1.1. Representación de los perfiles de los pozos de perforación minera.

En el proceso de explotación minera, uno de los trabajos realizados es la adquisición de información acerca de las características de las formaciones que se van atravesando durante la perforación de pozos. Estos datos obtenidos son de suma importancia para los especialistas mineros ya que les permite extraer información relevante acerca de las rocas que se van atravesando, las más comunes son:

⁵ Tipo de material que al ser cortado y pulido se puede usar como decoración y alumbramiento.

⁶ Es el estudio del crecimiento, la forma y el carácter geométrico de las formas cristalinas que adoptan los minerales bajo condiciones favorables.

- Definir los límites de las capas de roca.
- Definir los cambios en el tipo de roca, esta incluye:
 - Cuando se encuentra un acuífero en el pozo.
 - Distinguir esquisto⁷ arcilloso, piedra arenisca y caliza.
 - De las capas de carbón, delimitar profundidad, espesor y grado.
 - Rocas mineralizadas y su espesor.
- Relevar si el acuífero es roca porosa o fracturada.
- Encontrar fugas acuíferas de agua salada o dulce.

La estratigrafía es una rama de la geología que se encarga de la identificación, descripción, estudio e interpretación de las rocas sedimentarias estratificadas (Weller, 1960). Esta rama estudia las rocas teniendo en cuenta la secuencia temporal de estas y los materiales que la constituyen. Una de las áreas que recoge la estratigrafía es la litoestratigrafía la cual tiene el objetivo de caracterizar la litología de las diferentes sucesiones estratigráficas. De acuerdo con esto es que surgen los conceptos de perfil litológico, columna litológica y columna estratigráfica que representan gráficamente la estratigrafía generalizada de un sector (Dabrio y otros, 2003).

Un perfilado de pozo es una de las actividades más importantes en la minería, esta consiste en la toma y monitoreo de los perfiles o registros de pozos. Un perfil de pozo es una grabación contra profundidad de algunas de las características de las formaciones rocosas atravesadas, hechas con herramientas de medición en el hoyo del pozo (Dabrio y otros, 2003). Estos perfiles de los pozos son muy importantes, ya que a través de estos se miden un número de parámetros físicos relacionados con las propiedades geológicas de los estratos que se han penetrado. En otras palabras los perfiles de pozos constituyen una descripción de la roca perforada y de las características de los minerales que posee dicho pozo.

⁷ Roca de color negro azulado que se divide con facilidad en hojas.

La automatización del proceso de construcción de los perfiles de los pozos de perforación minera es uno de los aspectos fundamentales para todas las empresas mineras. Las gráficas generadas en este proceso son ampliamente utilizadas en operaciones como: análisis de datos, estudios comparativos, exploración, análisis de riesgos, entre muchas otras. Esta información geológica minera en su forma natural es difícil de analizar por los especialistas mineros debido principalmente a la considerable cantidad de datos acumulados. No poseer un software que cubra las necesidades para llevar a cabo esta tarea, al cual se le puedan hacer modificaciones para mejorarlo y adaptarlo de acuerdo a las necesidades del lugar donde se use, es sin duda alguna una seria desventaja para cualquier empresa dedicada a esta área del negocio.

Las aplicaciones con que cuenta actualmente el país para llevar a cabo la representación de los perfiles de los pozos de perforación son propietarias, por lo que están protegidas por una licencia electrónica, lo que supone un problema para las empresas que utilizan estas soluciones (Chávez y otros, 2012).

1.2 Análisis de otras soluciones existentes

Actualmente la minería se ha convertido en un negocio muy competitivo a nivel mundial por lo que se trabaja en pos de lograr una mayor eficiencia en este campo. Diversas alternativas ofrece la industria de software para el apoyo al trabajo minero, desde soluciones genéricas para la gestión de todas las áreas del negocio, hasta soluciones específicas de una rama determinada. Es importante destacar la finalidad del uso de estas herramientas y sistemas mineros que logran prácticamente el máximo del aprovechamiento de los yacimientos minerales. El principal problema de estos, es que son privativos e imponen una dependencia tecnológica propiciada por la necesidad de usarlos, lo que conlleva a pago de versiones, licencias, capacitación, y servicios, que atan de alguna forma a las empresas que lo utilizan con las compañías productoras de estos sistemas.

A continuación se describen algunas soluciones informáticas para la representación de las columnas de los pozos de perforación, se mencionan los elementos más importantes y de interés que puedan ofrecer un conjunto de ideas para la construcción de la herramienta que se tiene como propuesta de solución.

La herramienta *LogPlot 7* es ampliamente utilizada para aquellas empresas que se dedican a la extracción de minerales industriales como metales, carbón, piedra caliza o cualquier otro recurso mineral. Este producto de software posee un perfeccionado formato para la entrada de datos de litología, además cuenta con una representación mediante gráficas de barra de las columnas de los pozos de perforación,

así como la concentración de los diferentes minerales mediante gráficas de líneas, entre muchos otros elementos que brindan toda la información necesaria a los especialistas. Otro de los elementos principales con que cuenta este sistema es que proporciona una vista de todos los elementos en una sola página de manera que el usuario puede guardar la información como imágenes con extensiones JPG, BMP, PNG y otros.

LogPlot 7 también brinda la posibilidad al usuario de asignar texturas y colores a las barras que representan a las litologías y a las gráficas de líneas que representan a las concentraciones de los minerales, así como proporcionar una descripción de estos para un mejor entendimiento del usuario. De esta herramienta se puede mencionar que no tiene en cuenta la representación de la localización de los pozos en el terreno, ni la representación de la estructura litológica del terreno, que no es más que las columnas litológicas de un conjunto de pozos simultáneamente. Además tiene el importante inconveniente de estar concebida para usuarios que dominen el idioma inglés. En la figura 1 se puede observar varios de los elementos que tiene en cuenta este sistema para la representación de los perfiles de los pozos de perforación.

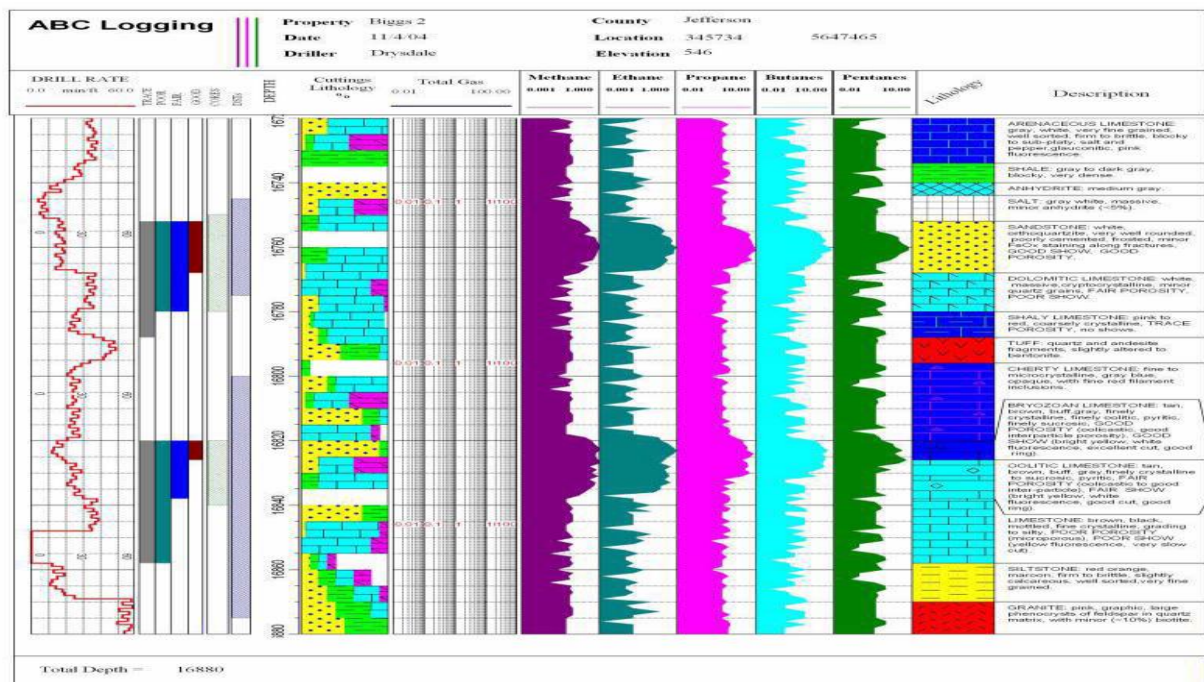


Fig. 1.-Herramienta LogPlot 7.

Una vez que el equipo de desarrollo interactuó directamente con la herramienta *LogPlot 7*, se tomaron algunos elementos que proporcionan ideas para la construcción de la herramienta propuesta, la forma de representar la litología de los pozos y la concentración de cada uno de los minerales es el principal aporte de esta a la investigación. También es de gran importancia la funcionalidad que brinda al usuario de exportar toda la información referente a un pozo de perforación en diferentes extensiones teniendo en cuenta la necesidad de los especialistas mineros.

Otra de las herramientas más usadas en la actualidad es el producto de representación de sondajes y registros de pozos *DownHole Explorer*. El programa es capaz de manejar casi cualquier tipo de datos, desde prácticamente cualquier fuente. Proporciona una amplia gama de formas de representar la información, además que brinda funcionalidades para personalizar estos datos donde se incluyen las texturas, el color, tamaño, fuente y estilo (Environmental Systems Research Institute, 2004).

DownHole Explorer también tiene un sistema de tablas en tiempo de ejecución para mostrar los registros de los pozos, lo que permite ahorrar tiempo en consultas a bases de datos u otras fuentes de información. Al igual que *LogPlot 7* la herramienta *DownHole Explorer*, no tiene en cuenta la representación de la localización de los pozos en el terreno, ni la representación de la estructura litológica del terreno. Aunque brinda un conjunto más amplio de funcionalidades que el sistema antes visto, esta omite algunas que son indispensables, como la de no representar la concentración de los minerales en gráficas de líneas y limitar al usuario el acercamiento y alejamiento de las gráficas para una mejor apreciación. En la figura 2 se puede apreciar una vista de esta aplicación informática (Environmental Systems Research Institute, 2004).

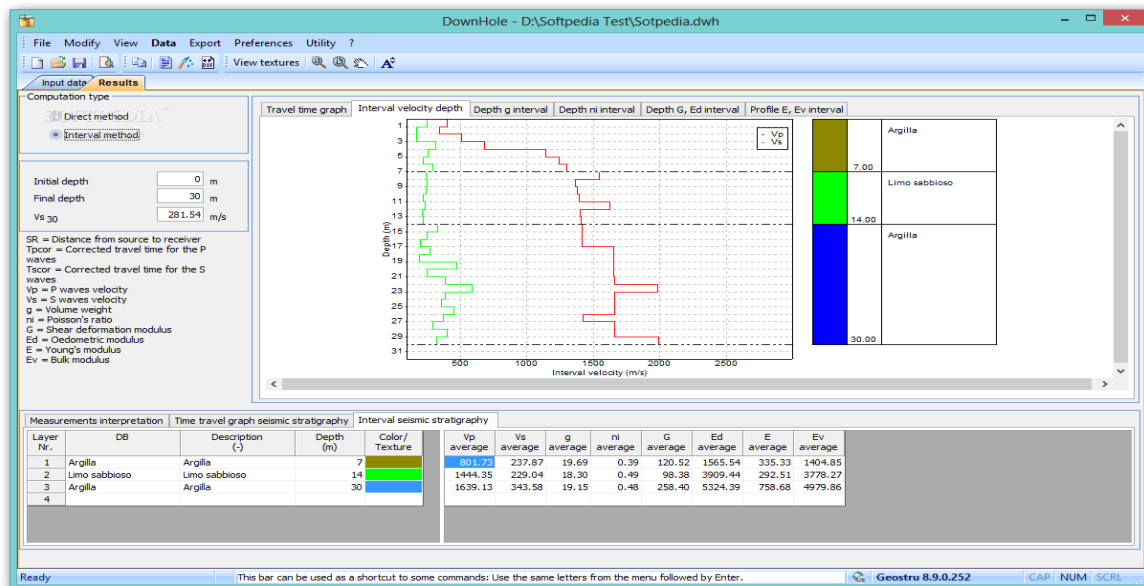


Fig. 2.-Herramienta *DownHole Explorer*.

Este sistema tiene la desventaja de no ser multiplataforma, pero proporciona un conjunto de ventajas y características beneficiosas que pueden ser utilizadas para la implementación de la herramienta a desarrollar. Dentro de los elementos que se tomaron se puede mencionar como uno de los más importantes, la funcionalidad de personalizar la información mostrada teniendo en cuenta la textura, color, fuente, estilo, tamaño entre otros, que permiten al usuario una interacción más completa con el software. Se considera que esta solución es capaz de importar diversas estructuras de datos debido a que brinda al usuario la posibilidad de relacionar los datos importados con los que espera la aplicación, una forma de importar similar va a ser implementada en la herramienta a desarrollar de manera que esta sea más genérica respecto a archivos con distintos formatos.

1.3 Lenguajes de Programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Está formado por un conjunto de símbolos, reglas sintácticas y reglas semánticas que definen su estructura y el significado de sus elementos y expresiones.

También se puede definir como un idioma artificial diseñado para ordenar tareas a realizar por máquinas como las computadoras, que puede usarse para crear programas que controlen el comportamiento físico y

lógico de la computadora. Dentro de los lenguajes de programación se encuentran los de bajo nivel y los de alto nivel que se caracterizan por expresar los algoritmos necesarios para la creación de programas informáticos de manera que facilitan la comunicación entre un humano y la computadora mediante signos convencionales cercanos al lenguaje natural. Algunos de los lenguajes de programación más empleados en la actualidad son *C++*, *Java*, *C#*, entre muchos otros.

1.3.1. C++ como lenguaje de programación

C++ es un lenguaje de programación de alto nivel considerado uno de los más potentes, diseñado a mediados de 1980 como extensión del lenguaje de programación C, abarca tres paradigmas de programación: la programación estructurada, genérica y orientada a objeto.

Algunas de las características principales del lenguaje de programación C++ son:

- Tiene un conjunto completo de instrucciones de control.
- Permite la agrupación de instrucciones.
- Incluye el concepto de puntero (variable que contiene la dirección de otra variable).
- Los argumentos de las funciones se transfieren por su valor.
- La entrada/salida no forma parte del lenguaje, sino que se proporciona a través de una biblioteca de funciones.
- Permite la separación de un programa en módulos que admiten compilación independiente.

Se selecciona el lenguaje de programación C++ para la construcción de la herramienta para la representación de las columnas de los pozos de perforación ya que es uno de los lenguajes más empleados en la actualidad que cuenta con gran cantidad de bibliografía, materiales didácticos y con la experiencia en el desarrollo de este lenguaje por parte del equipo de desarrollo. Este lenguaje estandarizado que puede compilarse en varias plataformas, es rápido en cuanto a tiempo de ejecución y es capaz de utilizar características de bajo nivel para realizar implementaciones óptimas logrando que los programas sean más compactos y rápidos.

1.4 Framework de desarrollo Qt

Es una amplia plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica en C++ que pueden operar en varias plataformas. Qt dispone de una

amplia gama de herramientas que facilitan la creación de formularios, botones y ventanas de diálogo con el uso del *mouse* (Thelin, 2007).

Ventajas del uso del *framework*⁸ de desarrollo Qt:

- Qt es completamente gratuito para aplicaciones de código abierto.
- Las herramientas, librerías y clases están disponibles para casi todas las plataformas, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará de la misma forma.
- Qt tiene una extensa librería con clases y herramientas para la creación de aplicaciones. Estas librerías y clases están bien documentadas (Thelin, 2007).

Se selecciona como *framework* de desarrollo Qt, ya que es multiplataforma, provee de una plataforma que facilita el trabajo con C++ que fue el lenguaje seleccionado para construir la herramienta. Qt cuenta con varias clases definidas por él, que enriquecen las librerías del C++ y además presenta algoritmos que facilitan la implementación del negocio minero, los que han sido utilizados en el desarrollo de aplicaciones de gran reconocimiento y prestigio a nivel mundial en el campo de la minería. Además Qt tiene amplia documentación sobre sus clases y librerías y el equipo de desarrollo tiene experiencia en el uso de este *framework* de desarrollo.

1.5 Biblioteca QCustomPlot

QCustomPlot es un *Widget*⁹ escrito en C++ sobre el *framework* de desarrollo Qt. Es una biblioteca totalmente documentada a través del sitio <http://www.qcustomplot.com/>, no posee dependencias y ofrece un alto rendimiento para aplicaciones de visualización en tiempo real. Se especializa en hacer trazados de gráficas y mapas en dos dimensiones.

⁸ Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

⁹ Es una abreviación de las palabras *window* y *gadget* (ventana y dispositivo, en inglés). Un *widget* es una mini-aplicación de ordenador que se presenta como una pequeña ventana o caja.

Esta biblioteca propone una gráfica con ejes de coordenadas (X, Y) en el cual se pueden insertar un amplio conjunto de funciones, estas abarcan disímiles variedades y formas que se pueden personalizar en cuanto a un conjunto de propiedades, entre ellas se encuentra la edición de los colores de borde y fondo de dichas gráficas representadas. Dichas gráficas permiten la inserción de textos, de leyendas para las funciones, de varios ejes de coordenadas, entre muchas otras funcionalidades que la convierten en una biblioteca que puede ser usada para muchos fines (Eichhammer, 2006).

QCustomPlot ofrece funcionalidades de interacción con el usuario, estas se pueden dividir en tres grupos principales (Eichhammer, 2006):

- Manipulación de rango arrastrando la rueda del *mouse*.
- La selección de funciones independientes en una gráfica haciendo uso del *mouse*.
- Se pueden recibir señales emitidas por el usuario haciendo uso del *mouse*.

La figura 3 muestra ejemplos de lo que se puede lograr con el uso de esta biblioteca.

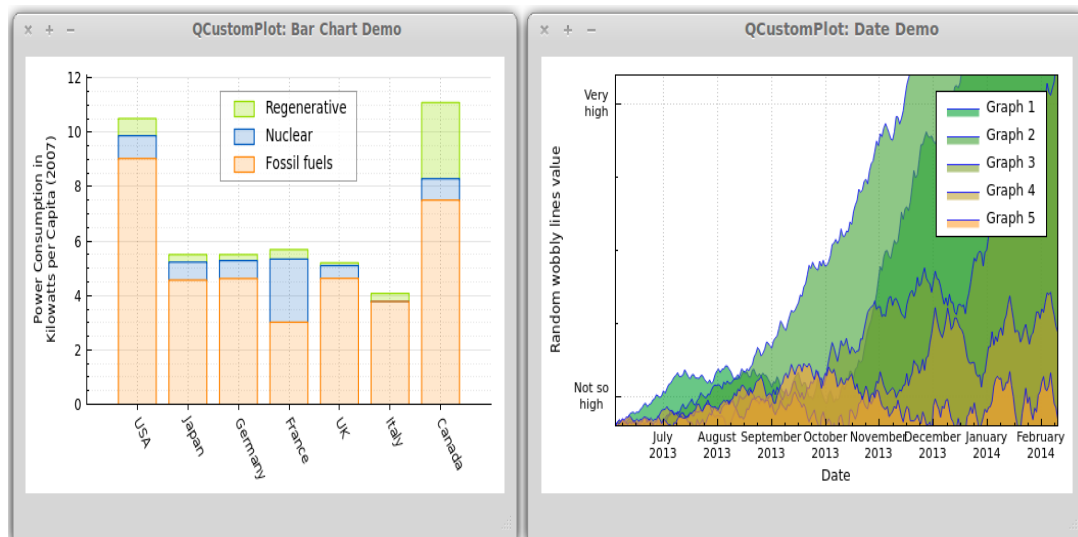


Fig. 3.-Ejemplos de la biblioteca *QCustomPlot*.

Esta biblioteca será usada por el equipo de desarrollo para representar las columnas litológicas de los pozos de perforación minera, así como las concentraciones de los minerales que van surgiendo durante la

perforación. Presenta un conjunto de funcionalidades y características que lo hacen ideal para dichas representaciones. Además permite exportar las representaciones a varios formatos como archivos PDF¹⁰ e imágenes de extensiones PNG, JPG y BMP, los archivos PDF generados pueden ser importados y editados con algunos de los editores conocidos para este tipo de archivos.

1.6 Sistemas Gestores de Bases de Datos

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas informáticos que permiten el almacenamiento, modificación y extracción de la información en una base de datos¹¹, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los SGBD 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 (Valderrey, 2011).

Un SGBD tiene las siguientes características (Valderrey, 2011):

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de los datos.
- Control centralizado.

Actualmente existen varios Sistemas Gestores de Bases de Datos, algunos de los más comunes son *PostgreSQL*, *Microsoft SQL Server*, *MySQL*, *Oracle*, entre otros.

¹⁰ Sigla del inglés de *portable document format*.

¹¹ Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

1.6.1. *PostgreSQL como Sistema Gestor de Bases de Datos*

PostgreSQL es un sistema gestor de bases de datos relacional orientado a objeto y con su código fuente disponible libremente, publicado bajo la licencia BSD (por sus siglas en inglés *Berkeley Software Distribution*). Como muchos otros proyectos de código abierto, es dirigido por una comunidad de desarrolladores. *PostgreSQL* utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema (Momjian, 2000).

A continuación, algunas de las características más importantes y soportadas por *PostgreSQL* (Momjian, 2000):

- Es una base de datos 100% *ACID*¹².
- Soporta distintos tipos de datos y permite la creación de tipos propios.
- Múltiples métodos de autenticación.
- Es multiplataforma.
- Soporta almacenamiento de objetos grandes como gráficos, videos, sonidos, entre otros.

Se selecciona como gestor de base de datos *PostgreSQL* debido a que por la arquitectura de diseño del mismo posee gran escalabilidad. Es capaz de ajustarse a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Además, tiene ciertas características orientadas a objetos. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser *Oracle*. Este SGBD posee opciones de seguridad flexibles en un extenso conjunto de protocolos de seguridad y opciones de configuración como rasgos internos, que ayudan a tener el control sobre quiénes y que accede a los datos dentro de la base de datos. Lo más importante es que debido a que el mismo se licencia bajo *BSD*¹³ se

¹² Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

¹³ Siglas en inglés de *Berkeley Software Distribution*.

inscribe en la definición de software libre, es uno de los SGBD más usados en la universidad, el equipo de desarrollo tiene experiencia en el uso de este sistema y se cuenta con abundante bibliografía y capacitación sobre este sistema.

1.7 Metodologías de desarrollo de software

Una metodología es un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software, además de constituir una asistencia fundamental cuando se da mantenimiento o actualización a un software implementado (Pressman, 2002).

Las metodologías de desarrollo de software surgen como alternativa al problema de lo difícil que resulta la construcción de un producto informático. Estas imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Actualmente existen numerosas propuestas metodológicas donde cada una de ellas tiene ventajas y desventajas, seleccionar la adecuada para obtener los resultados óptimos es uno de los problemas que enfrenta un equipo de desarrollo.

Las metodologías de desarrollo se pueden dividir en dos grandes grupos de acuerdo con sus características y objetivos que persiguen: ágiles y tradicionales. Algunas de las más usadas son: El Proceso Unificado de Desarrollo de Software (*RUP*), Programación Extrema (*XP*), Proceso Unificado Abierto (*OpenUP*), entre muchas otras.

1.7.1. Programación Extrema (XP) como metodología de desarrollo de software

Las metodologías ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración (Gallo, y otros, 2012).

A continuación se muestran un conjunto de principios y características que son comunes en los procesos ágiles de desarrollo de software (Beck, y otros, 2000):

- La principal prioridad es la entrega temprana y continua de software de valor.

- Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblan al cambio como ventaja competitiva.
- Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- El software que funciona es la principal medida del progreso.
- Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica enaltece la agilidad.
- La simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial.
- En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

Programación Extrema (*XP*) es una de las metodologías más exitosas en la actualidad y forma parte del grupo de las metodologías ágiles, está más orientada a la generación de código con ciclos muy cortos de desarrollo, se dirige a equipos de desarrollo pequeños y hace especial hincapié en aspectos humanos asociados al trabajo en equipo. *XP* es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y divertida para desarrollar software (Pressman, 2002).

XP es una metodología única y compacta, resultado de una selección de aquellas prácticas que se han considerado mejores y han profundizado en sus relaciones y en cómo se refuerzan unas con otras. Aunque no está basado en nuevos principios el resultado sí es una nueva manera de ver el desarrollo de software, los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

De acuerdo a los elementos antes vistos se selecciona la metodología de desarrollo *XP*, que se emplea en proyectos de corto plazo y es usada principalmente para equipos pequeños. Esta metodología integra de una forma muy efectiva cada una de las prácticas de desarrollo que propone y en cada una de ellas está muy bien definido el porqué de su utilización y los beneficios que esta representa. Ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

1.8 Lenguaje de Modelado

Actualmente es imposible imaginarse el desarrollo de software sin antes pasar por una etapa de modelado. La falta de estandarización en la forma de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores. Algunos de los lenguajes de modelado más conocidos son: la Notación de Modelado de Procesos de Negocio (*BPMM*), el lenguaje de Modelado de Sistemas (*SysML*) y Lenguaje Unificado de Modelado, conocido como *UML* por sus siglas en inglés.

1.8.1. **Lenguaje Unificado de Modelado (UML)**

El Lenguaje Unificado de Modelado (*UML* siglas de *Unified Modeling Language*) es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Sus creadores pretendieron unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas estandarizándolas.

UML es el estándar de la industria en la actualidad debido a un conjunto de rasgos principales dentro de los que se pueden mencionar (Rumbaugh, y otros, 2000):

- Permite modelar sistemas utilizando técnicas orientadas a objetos.
- Es un lenguaje muy expresivo que cubre las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Ampliamente utilizado por la industria del software.

- Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- Modela estructuras complejas.
- Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración, que sirve para evaluar el estado de las máquinas.

Se tomó la decisión de utilizar *UML* para el modelado de la herramienta para la representación de los perfiles de los pozos de perforación ya que es de fácil comprensión para su uso por parte del equipo de desarrollo y es menos engorroso realizar cambios una vez que se haya comenzado a desarrollar el software, lo que se complementa con la metodología escogida que tiene la característica de adaptarse rápidamente a constantes cambios. El uso de este lenguaje de modelado facilita el entendimiento de los procesos realizados por el sistema para el desarrollador y a su vez facilita la comunicación entre los desarrolladores, lo que permite el ahorro de tiempo en el desarrollo de software.

1.9 Herramienta CASE

Se puede definir a una herramienta CASE (en español, ingeniería de software asistida por computadora, en inglés, *Computer Aided Software Engineering*) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante los pasos del ciclo de vida del desarrollo de un software. (Kendall, y otros, 2005). Entre los principales objetivos tiene:

- Mejorar la productividad en el proceso de desarrollo del software.
- Aumentar el nivel de calidad del sistema construido.
- Reducir el tiempo y coste de desarrollo.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayudar a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestionar de forma global todas las fases de desarrollo del software con una misma herramienta.

- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Algunas de las herramientas *CASE* más usadas en el mundo para el desarrollo de software son *Rational Rose*, *Enterprise Architect* y *Visual Paradigm*.

1.9.1. Visual Paradigm como herramienta CASE

Visual Paradigm es una herramienta *CASE* que utiliza *UML* como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios (Rumbaugh, y otros, 2000).

Es considerada una de las herramientas más completas y fáciles de usar, es multiplataforma y proporciona facilidades de interoperabilidad con otras aplicaciones. Es una herramienta poderosa y colaborativa ya que permite la realización de ingeniería directa e inversa y a su vez soporta múltiples usuarios trabajando sobre el mismo proyecto. Además mejora la productividad, aumenta la calidad y reduce el tiempo en el desarrollo y mantenimiento del software.

Algunas de las ventajas que posee la herramienta *Visual Paradigm* son:

- Un entorno de creación de diagramas para *UML*.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanecen sincronizados en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, en varios idiomas.
- Es multiplataforma.

Se selecciona *Visual Paradigm* como herramienta *CASE* ya que posee un conjunto de características y ventajas que la convierten en la que más se ajusta para el desarrollo de la herramienta que se tiene como propuesta de solución. Esta herramienta es capaz de generar código en C++ y utiliza *UML* lo que permite una interoperabilidad entre las herramientas seleccionadas como lenguaje de programación y modelado respectivamente. A su vez posibilita la representación gráfica de los diagramas, como por ejemplo: secuencia, casos de uso, clases, actividad, estado, entre otros. Es una tecnología libre, multiplataforma, fácil de instalar y actualizar que se encuentra disponible en español. Además se tuvo en cuenta que el equipo de desarrollo tiene experiencia en el uso de esta herramienta, que la universidad posee la licencia para el empleo de esta en sus proyectos productivos y que se cuenta con una versión nativa de *Linux*.

1.10 Conclusiones

Las herramientas con que cuenta el país para representar los perfiles de los pozos de perforación minera son privativas por lo que no resuelven totalmente el problema debido a varias limitaciones que impiden su utilización. Por otro lado las tecnologías, herramientas, lenguajes y *framework* de desarrollo seleccionados son adecuadas para el desarrollo de la herramienta que se tiene como propuesta de solución y simultáneamente obedecen a las políticas de software libre que impulsa la universidad y el país. Por último quedan creadas las condiciones para dar comienzo a la siguiente fase de la investigación.

CAPÍTULO II: Características de la herramienta para la representación de los perfiles de los pozos de perforación minera

En el Capítulo I fueron identificadas algunas aplicaciones que servían de apoyo a las empresas mineras en la representación de los perfiles de los pozos de perforación, en el presente capítulo se abordarán las vías para desarrollar una herramienta de carácter nacional que sirva de apoyo a las empresas nacionales.

Se realiza una descripción de las principales definiciones asociadas al dominio del área que se desea informatizar, se desarrollan las primeras fases del ciclo de vida de la metodología *XP*: planificación y diseño. Son creadas las historias de usuarios y se describen las tarjetas CRC¹⁴ para un mejor entendimiento del sistema.

2.1. Modelo de dominio

El modelo de dominio captura los tipos de objetos más importantes que existen, teniendo como características que permite ayudar a los desarrolladores a utilizar un vocabulario común, para lograr una mejor comprensión del contexto en que se realiza el sistema.

El modelo de dominio es una representación de los conceptos (objetos) significativos en el dominio del área que se desea informatizar y las relaciones existentes entre estos. En la figura 4 se muestra el modelo de dominio.

¹⁴ Cargo o Clase, Responsabilidad y Colaboración.

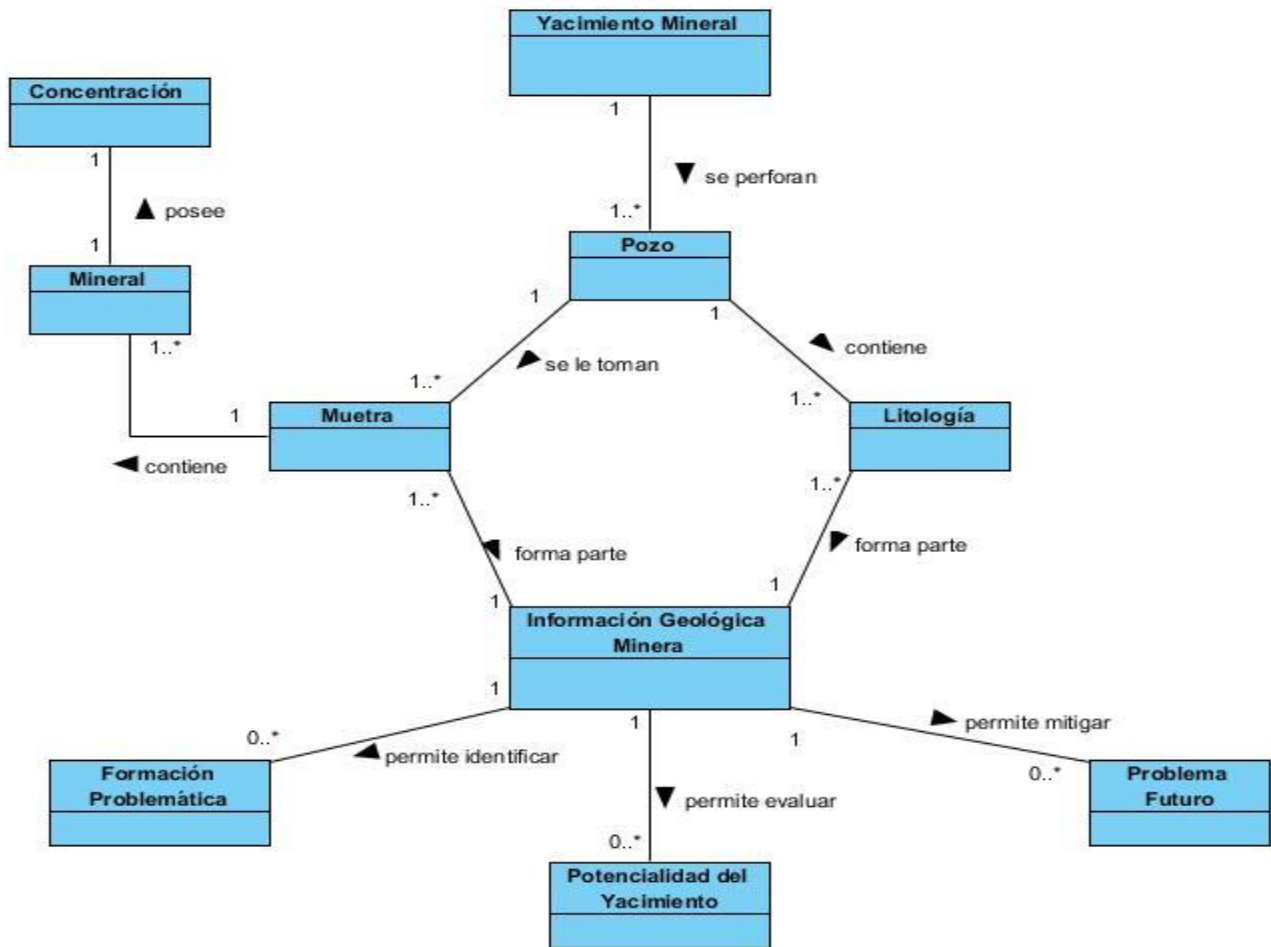


Fig. 4.-Modelo de Dominio.

2.1.1. Descripción del modelo de dominio

Los yacimientos minerales son acumulaciones naturales de minerales que pueden ser objeto de extracción y explotación industriales con fines económicos. En estos yacimientos se perforan un conjunto de pozos a los cuales se le toman muestras (Assays), de las que se conocen los minerales que posee y la concentración de cada uno de ellos. También se conoce las litologías que posee el pozo, es decir los tipos de roca que se encuentran en toda su trayectoria.

Estos datos en su conjunto constituyen la información geológica minera del pozo, la cual es ampliamente usada para el estudio y análisis de los yacimientos minerales por los especialistas mineros. El principal uso de esta información es que permite a los perforadores identificar formaciones problemáticas, mitigar

posibles problemas futuros en otras perforaciones en lugares equivalentes y ofrece la posibilidad de clasificar y evaluar la potencialidad del yacimiento.

2.2. Descripción de la solución

La herramienta para la representación de los perfiles de los pozos de perforación minera tiene que representar tres componentes fundamentales a partir de la información geológica minera de un pozo, estos son las diferentes litologías que este presenta en forma de gráfica de barra, las concentraciones de cada uno de los minerales que este posee en forma de gráfica de línea y la estructura litológica del terreno, que no es más que la representación de un conjunto de columnas litológicas en un sector determinado.

Esta herramienta obtiene la información geológica minera a través de bases de datos y de ficheros existentes. La aplicación también debe permitir gestionar y personalizar estos dos componentes de acuerdo a las necesidades del usuario y exportar documentos *PDF* e imágenes de extensiones *BMP*, *JPG* y *PNG*.

2.3. Fase de Diseño

La metodología *XP* define una serie de actividades para guiar el proceso de desarrollo de un software, entre las más importantes figuran la definición de las historias de usuarios, plan de reserva del producto, plan de iteraciones y las tarjetas CRC.

2.3.1. Historias de usuarios

Las Historias de Usuario (HU) son la técnica utilizada para especificar los requisitos del software en la metodología *XP*. Se trata de tarjetas en las cuales se describe brevemente las características que el sistema debe poseer, de estas surgen los requisitos funcionales y no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en semanas (Letelier, y otros, 2010).

Las HU tienen como función principal describir las tareas a realizar por el sistema. Estas HU son escritas en lenguaje natural y de forma concisa, además las historias de usuario solamente proporcionarán los detalles sobre la estimación del riesgo y cuánto tiempo conllevará la implementación de dicha historia de usuario. El tiempo se estimará en semanas por el equipo de desarrollo de acuerdo a experiencias anteriores y a la complejidad de la historia de usuario que se esté estimando. Debido a la limitada información que posee una historia de usuario principalmente la poca información sobre las implicaciones técnicas de su implementación, el tiempo que se estima es solo una aproximación para que los desarrolladores tengan una idea de la demora total de una determinada iteración.

A continuación se detallan algunas de las historias de usuario consideradas las más importantes para el negocio, en el Anexo I y en los artefactos generados se encuentran las restantes historias de usuario generadas:

Tabla 1. Historia de Usuario #1.

| Historia de Usuario | |
|---|--|
| Número: 1 | Nombre de la Historia de Usuario: Conectar aplicación con la base de datos. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 1 |
| Prioridad en negocio: Muy Alta. | Puntos estimados(semanas): 0.4 |
| Riesgo en desarrollo: Media. | Puntos reales(semanas): 0.3 |
| <p>Descripción: El geólogo se conecta a través de la aplicación con la base de datos. Para conectarse con la base de datos el geólogo debe brindar la siguiente información al sistema:</p> <ul style="list-style-type: none"> - Servidor - Puerto - Usuario - Contraseña <p>Una vez que se establezca la conexión se muestran las bases de datos existentes y el geólogo selecciona una y la aplicación carga los datos de dicha base de datos.</p> <p>De la base de datos son necesarios para el funcionamiento de la aplicación los siguientes datos:</p> | |

- Nombre del pozo.
- Coordenadas del pozo.
- Assay (muestras), de cada muestra son necesarios los siguientes datos:
 - From (desde).
 - To (hasta).
 - Litologías.
 - Concentración de cada uno de los minerales.

Observaciones: Para que el geólogo pueda conectar la aplicación con la base de datos debe introducir correctamente los datos para la conexión y debe existir al menos una base de datos.

Prototipo de interfaces

Conectarse a base de datos.

Servidor:

Puerto:

Usuario:

Contraseña:

BD:

Tabla 2. Historia de Usuario #2.

| Historia de Usuario | |
|--|---|
| Número: 2 | Nombre de la Historia de Usuario: Representar la columna litológica. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |

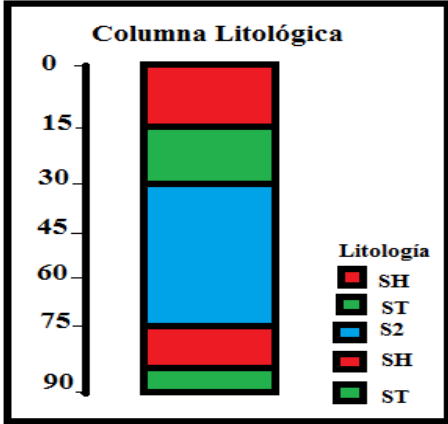
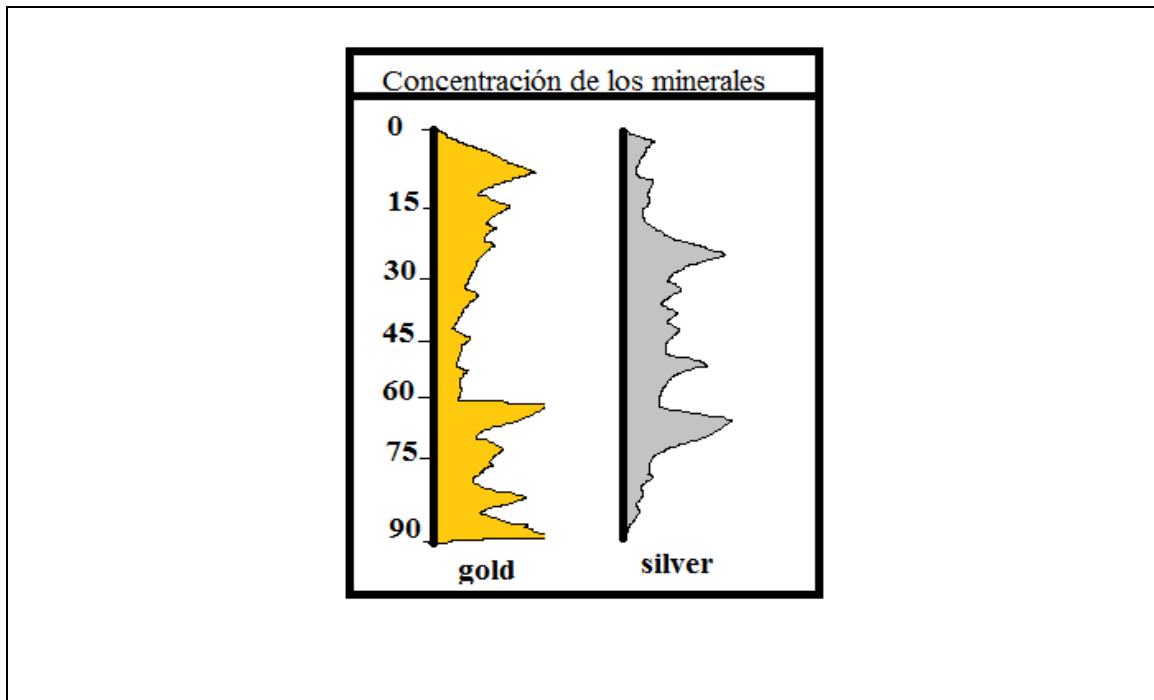
| | |
|--|---------------------------------------|
| Usuario: Geólogo. | Iteración asignada: 1 |
| Prioridad en negocio: Muy Alta. | Puntos estimados(semanas): 2.0 |
| Riesgo en desarrollo: Muy Alta. | Puntos reales(semanas): 2.2 |
| <p>Descripción: El geólogo una vez conectada la aplicación con la base de datos podrá visualizar la columna litológica de un pozo seleccionado previamente. Cada pozo está dividido por tramos con diferentes tamaños y cada uno de estos tramos presenta una litología. El pozo debe ser representado en una gráfica con ejes de coordenadas Y para representar la profundidad del pozo. Dicho pozo debe representarse en forma de columna colocando las diferentes litologías en forma de barras una sobre otra teniendo en cuenta el tamaño del tramo de dicha litología. También se debe mostrar el código de las litologías en el respectivo orden en que fueron apareciendo en los tramos de cada pozo.</p> | |
| <p>Observaciones: Una vez conectada a la base de datos se debe marcar un pozo aleatoriamente y mostrar su respectiva columna litológica y el código de las litologías de dicho pozo.</p> | |
| <p>Prototipo de interfaces</p> <div style="text-align: center;">  <p>The figure is a vertical stacked bar chart titled "Columna Litológica". The vertical axis (Y-axis) represents depth in meters, with markings at 0, 15, 30, 45, 60, 75, and 90. The horizontal axis (X-axis) represents the width of the well column. The bar is composed of several segments of different colors representing different lithologies. From top to bottom, the segments are: a red segment (SH) from 0 to approximately 12 meters; a green segment (ST) from 12 to approximately 22 meters; a blue segment (S2) from 22 to approximately 65 meters; a red segment (SH) from 65 to approximately 75 meters; and a green segment (ST) from 75 to 90 meters. A legend on the right side of the chart, titled "Litologia", shows color-coded boxes for SH (red), ST (green), and S2 (blue).</p> </div> | |

Tabla 3. Historia de Usuario #3.

| Historia de Usuario | |
|---|---|
| Número: 3 | Nombre de la Historia de Usuario: Representar la concentración de los minerales. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 1 |
| Prioridad en negocio: Muy Alta. | Puntos estimados(semanas): 2.0 |
| Riesgo en desarrollo: Muy Alta. | Puntos reales(semanas): 1.6 |
| <p>Descripción: El geólogo una vez conectada la aplicación a la base de datos podrá visualizar la concentración de todos los minerales que contiene un pozo seleccionado previamente. Cada pozo está dividido por intervalos con diferentes tamaños y cada uno de estos intervalos presenta la concentración de todos los minerales que contiene el pozo. La concentración de los minerales debe ser representado en una gráfica de eje de coordenada X de acuerdo a los intervalos que posea el pozo. Dichas concentraciones se deben representar en una gráfica de línea de manera que muestre las alteraciones que sufren los minerales en los diferentes intervalos de todo el recorrido del pozo.</p> | |
| <p>Observaciones: Una vez conectada a la base de datos se debe marcar un pozo aleatoriamente y mostrar la concentración de todos los minerales que posee dicho pozo de la forma descrita. Se debe tener en cuenta que los intervalos del eje de coordenada de la gráfica de la columna litológica (Historia de usuario # 2) deben coincidir con los intervalos del eje de coordenada de la gráfica de las concentraciones de los minerales.</p> | |
| Prototipo de interfaces | |



2.3.2. Lista de reserva de producto

La Lista de Reserva de Producto (LRP) es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Los posibles elementos de esta lista son requerimientos técnicos y del negocio, funciones, y actualizaciones tecnológicas requeridas. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo se suelen conocer los más importantes que casi siempre son más que suficientes para una iteración (*sprint*), un *sprint* es un ciclo iterativo en el cual se desarrolla o mejora una funcionalidad para producir nuevos incrementos.

A continuación se muestra un fragmento de la lista de reserva del producto que contiene los requisitos de mayor prioridad para el negocio y que serán implementados en la primera iteración, en los artefactos generados se encuentra la LRP generada en la construcción de la herramienta:

Tabla 4. Fragmento de LRP.

| # | Descripción | Estimación (Semanas) | Estimado por |
|----------------------------|-------------|-------------------------|--------------|
| Prioridad: Muy Alta | | | |

| | | | |
|------------------------|--|-----|----------|
| 1 | Conectar aplicación con la base de datos. | 0.4 | Analista |
| 2 | Representar la columna litológica. | 2.0 | Analista |
| 3 | Representar la concentración de los minerales. | 2.0 | Analista |
| 4 | Mostrar los nombres de los pozos. | 0.3 | Analista |
| 5 | Representar columna litológica y concentración de los minerales de un pozo seleccionado. | 0.4 | Analista |
| 6 | Mostrar los tipos de litología de un pozo y los minerales existentes. | 0.5 | Analista |
| 7 | Representar estructura litológica del terreno. | 0.5 | Analista |
| Prioridad: Alta | | | |
| 8 | Editar color a las barras litológicas. | 0.2 | Analista |
| 9 | Aplicar textura a las barras litológicas. | 0.5 | Analista |
| 10 | Editar color de borde de la gráfica de la concentración. | 0.2 | Analista |
| 11 | Editar color de fondo de la gráfica de la concentración. | 0.2 | Analista |
| 12 | Cambiar el estilo de gráfica de la concentración a barra. | 0.4 | Analista |
| 13 | Cambiar el estilo de gráfica de la concentración a línea. | 0.4 | Analista |
| 14 | Asignar estilo de dispersión a las gráficas de concentración de estilo línea. | 0.2 | Analista |
| 15 | Mostrar la longitud de un pozo. | 0.2 | Analista |
| 16 | Exportar como PDF. | 0.5 | Analista |

2.3.3. Plan de iteraciones

El plan de iteraciones o plan *releases* permite definir el número de iteraciones en que se desarrollará el sistema de modo que se puedan definir con exactitud las entregas inmediatas y la entrega final. Este artefacto posee una estrecha relación con las historias de usuario anteriormente detalladas, ya que se tiene en cuenta la prioridad definida para cada una de estas historias de usuarios y se colocan en un orden según su prioridad. Al definir las prioridades antes vistas se arribó a la siguiente planificación, la

cual incluye tres iteraciones en donde se implementan en cada iteración las historias de usuarios con más prioridades para el negocio:

Tabla 5. Plan de iteraciones.

| No | Descripción de la iteración | Historias a implementar | Duración total(sem anas) |
|----|---|-------------------------|-----------------------------|
| 1 | Se desarrollan las Historias de Usuario más importantes para el negocio, de manera que se pueda tener una iteración con los elementos indispensables para este. Esto incluye la representación de las columnas de los pozos de perforación y sus respectivas concentraciones de los diferentes minerales. | 1,2,3,4,7,10 | 6.6 |
| 2 | Se desarrollan las funcionalidades que permiten la gestión a los componentes fundamentales, así como las distintas exportaciones a imágenes y la importación y exportación de ficheros. | 5,6,8,9,11,12,13 | 8.2 |
| 3 | Se desarrollan las funcionalidades de menos prioridad para el negocio referente a la gestión de las propiedades de las gráficas usadas para la representación. | 14,15,16,17,18 | 5.0 |

2.3.4. Tarjetas CRC

La metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniéndose después de cada iteración un producto funcional. Para poder diseñar el sistema como un equipo se debe cumplir con tres principios: Cargo o Clase, Responsabilidad y Colaboración (CRC).

Las tarjetas CRC permiten que el equipo completo contribuya en la tarea de diseño, permitiendo al programador centrarse y apreciar el desarrollo orientado a objetos. Debido a la facilidad y entendimiento de su uso son utilizadas para el diseño de la herramienta que se tiene como propuesta de solución.

A continuación se muestran las dos tarjetas CRC más importantes generadas en la construcción de la aplicación, en los artefactos generados se encuentran las restantes tarjetas CRC diseñadas:

Tabla 6. Tarjeta CRC# 1.

| Tarjeta CRC | | #1 |
|--|---|----|
| Clase: CoreApplication | | |
| Responsabilidades | Colaboraciones | |
| Iniciar componentes de representación. Cargar componentes de representación. Inicializar los datos. Importar datos desde fichero. Cargar datos desde base de datos. Cargar datos desde base de datos local. | SaveWidget DatabaseEntities DialogLoader SqliteHandler PsqlHandler MetroDatabase OpenWidget AssayImport LithologyImport CollarImport | |

Tabla 7. Tarjeta CRC# 2.

| Tarjeta CRC | | #2 |
|---|--|----|
| Clase: CoreView | | |
| Responsabilidades | Colaboraciones | |
| Iniciar componentes visuales. Borrar todos los componentes visuales. Inicializar gráficas de representación. Cargar nombres de los pozos. Cargar propiedades de las gráficas. Borrar gráficas de representación. | Customplot FileWidget MetroWidget AppStyle AxisRect DrillHole | |

| | |
|--|--------------------|
| <p>Agregar datos a las gráficas.</p> <p>Crear acciones del menú.</p> <p>Crear pestañas.</p> <p>Actualizar cambios de propiedades.</p> <p>Crear menú para exportar imagen.</p> <p>Exportar a imagen.</p> <p>Salvar propiedades de un componente.</p> <p>Abrir archivo.</p> <p>Representar pozo seleccionado.</p> <p>Cambiar de componente de visualización.</p> <p>Cargar propiedades de un componente.</p> | <p>ColorLegend</p> |
|--|--------------------|

2.4. Arquitectura base de la aplicación

La arquitectura es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan, y la estructura de datos que utilizan los componentes. En un sentido más amplio, sin embargo, los componentes pueden generalizarse para representar elementos importantes del sistema y sus interacciones (Pressman, 2002).

La arquitectura de software no es más que un conjunto de patrones que proporcionan un marco para guiar la construcción de un software, es considerado el nivel más alto en el diseño de la arquitectura que establecen la estructura, funcionamiento e interacción entre las partes de software. La estructura de un sistema se constituye de componentes - módulos o piezas de códigos, estos componentes se organizan de acuerdo a ciertos criterios, que representan decisiones de diseño.

2.4.1. *Patrones de arquitectura*

El modelo Cliente/Servidor es una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información. En este modelo las transacciones se dividen en procesos independientes que cooperan mutuamente para intercambiar información, servicios o recursos. Se denomina cliente al proceso que solicita los recursos y servidor al proceso que responde a estas solicitudes.

La arquitectura empleada para el desarrollo de la aplicación es la arquitectura Cliente/Servidor que básicamente consiste en un cliente que realiza peticiones, y un servidor que da respuesta a estas. Esta arquitectura se caracteriza en la relación de los clientes con el usuario y un servidor que interactúa con los recursos a compartir.

La organización de los componentes del sistema se diseñó en tres capas, capa de presentación (vista), capa de negocio (lógica) y la capa de información (modelo). El estilo arquitectural en capas se basa en una distribución jerárquica de las responsabilidades para proporcionar una división efectiva de los problemas a resolver (Lorente, y otros, 2010).

La arquitectura en capas tiene muchas ventajas, una de ellas es que el diseño claramente define la separación entre la funcionalidad de cada capa, lo que facilita que en caso que ocurra algún cambio, sólo se tendrían que realizar las correcciones necesarias en el nivel requerido sin tener que revisar código de otros niveles. Por otra parte las capas superiores como la de presentación envía comandos a las capas inferiores como la de negocios y la de datos. Las capas inferiores no tienen ninguna dependencia con las superiores, permitiéndoles ser reutilizables en otros escenarios. La figura 5 muestra cómo quedan distribuidos los paquetes en la construcción de la aplicación. El paquete *CoreApplication* que se encuentra en la capa de negocio contiene la clase controladora, esta obtiene los datos de los paquetes *ORM* e *IO* que se encuentran en la capa de información y son los encargados de acceder a los datos. En la capa de presentación se encuentra el paquete *CoreView* que es construida a través de la clase controladora utilizando los datos obtenidos en la capa de información.

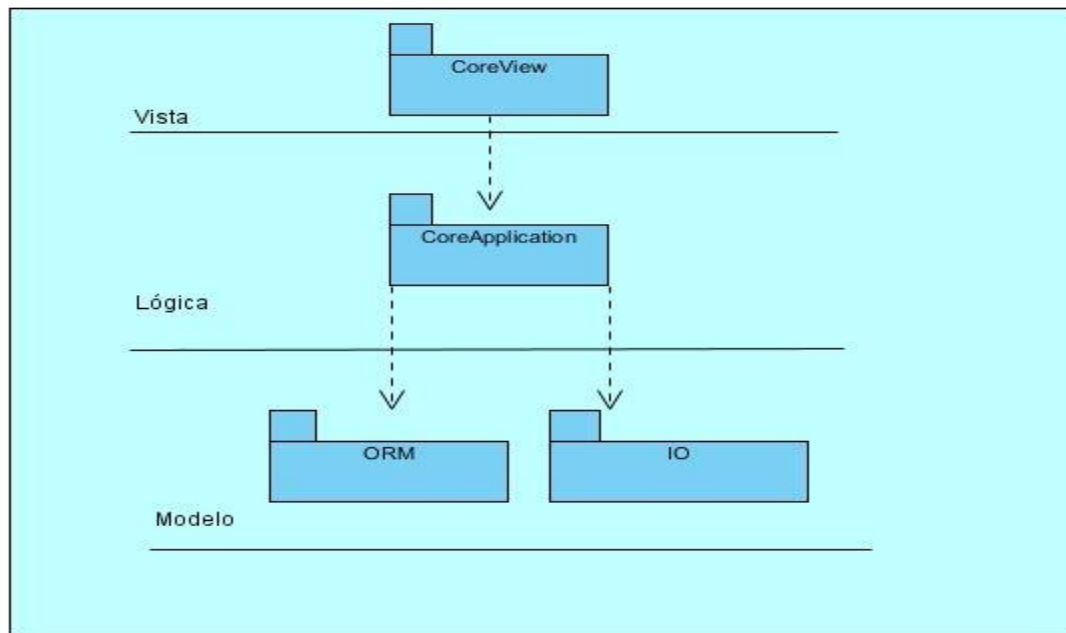


Fig. 5. -Diseño de la arquitectura.

2.4.2. Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia. En el desarrollo de multitud de aplicaciones hay problemas de diseños que se repiten o que son análogos, es decir, que responden a cierto patrón. Con el uso de patrones los diseños serán mucho más flexibles, modulares y reutilizables.

Existen un conjunto de familias de patrones de diseño, entre ellos se puede mencionar los patrones *GRASP*¹⁵ y los patrones *GOF*¹⁶. Los patrones *GRASP* contienen a los patrones Experto, Controlador, Bajo Acoplamiento, Alta Cohesión y Creador que describen los principios fundamentales de la asignación de responsabilidades a objetos. Por su parte los patrones *GOF* describen un conjunto de patrones que se agrupan en tres categorías: Creacionales, Estructurales y de Comportamiento.

¹⁵ Patrones de *Software* para la asignación General de Responsabilidad.

¹⁶ Abreviación del grupo *Gang of Four*.

En el presente trabajo se utilizaron los siguientes patrones GRASP:

Creador: Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Este patrón se puede evidenciar en la figura 6 donde la clase *DrillHole* posee una lista de objetos de la clase *Interval*.

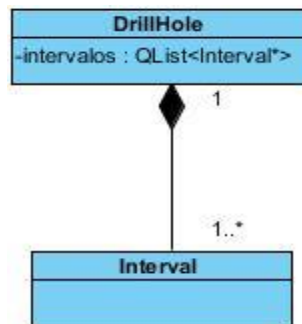


Fig. 6.-Patrón Creador.

Controlador: Para la aplicación de este patrón se utiliza una clase controladora que es la encargada de controlar los eventos del sistema, la clase controladora debería controlar todos los eventos correspondientes a una misma historia de usuario. La clase controladora garantiza que los procesos no sean manejados por la interfaz, ya que funciona como intermediaria entre esta y las clases donde reside la lógica de la aplicación. En la figura 7 se muestra la utilización de este patrón, la clase *CoreApplication* controla todos los eventos que se producen al importar desde un fichero, ya sean los nombres de los pozos, las litologías o las muestras.

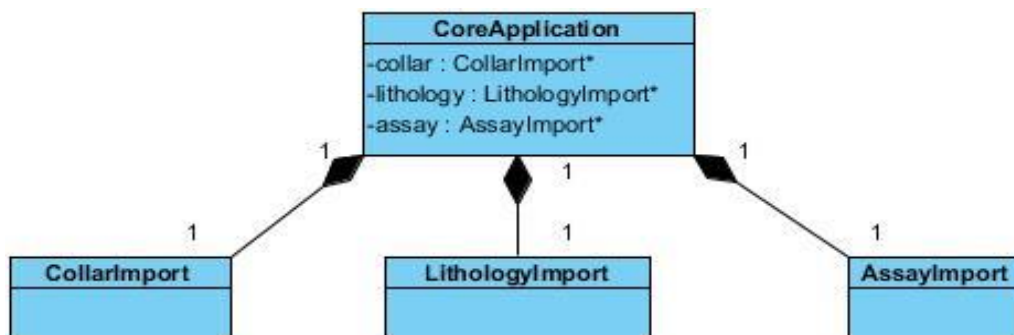


Fig. 7.-Patrón Controlador.

Alta Cohesión y Bajo Acoplamiento: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme. Con el uso de este patrón se mejoran la claridad y la facilidad con que se entiende el diseño. Una clase con bajo acoplamiento no depende de muchas clases. Con el uso de este patrón los componentes no se afectan por cambios en otros componentes, son fáciles de entender y de reutilizar logrando un diseño más modular. Estos dos patrones están estrechamente relacionados, ya que una alta cohesión generalmente implica un bajo acoplamiento.

En el presente trabajo se usaron los siguientes patrones *GOF*:

Patrones de comportamiento: Plantean la interacción y cooperación entre las clases. Esta clasificación de patrones estudia las relaciones entre las llamadas de los diferentes objetos e incluye un conjunto de patrones como el *State* (Estado), *Memento* (Recuerdo), *Observer* (Observador), entre otros. El patrón Observador define una dependencia entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen de manera automática todos los objetos que dependen de él.

Este patrón fue usado en abundancia ya que *Qt* lo trae por defecto, utiliza el mecanismo de la interacción con las clases visuales mediante las señales y *slots*, donde de forma asincrónica los objetos que están suscritos a esta señal son informados del cambio de estado.

Patrones de creación: Son los encargados de crear instancias de objetos. Abstraen la forma en que se crean los objetos, permitiendo tratar las clases a crear de forma genérica. Entre los patrones que incluye se encuentran *Singleton* (Instancia Única), *Factory Method* (Método de Fabricación), *Builder* (Constructor virtual), entre otros. El patrón de instancia única fue utilizado en la implementación de la herramienta, el mismo garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es importante el uso de este patrón ya que es necesario una única instancia de los elementos que van a ser representado, ya sea que los datos se obtengan de un archivo importado, de una base de datos o de un fichero local. En la figura 8 se puede apreciar la clase *CoreApplication* donde se tiene una instancia de la misma y las demás clases tienen un acceso a esta en todo momento.

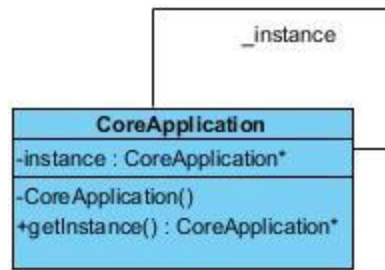


Fig. 8.-Patrón instancia única.

D-pointer

En la construcción de la herramienta se utiliza *d-pointer* que no es más que un patrón de diseño donde los detalles de la implementación de una librería pueden ser ocultados de sus usuarios y los cambios en la implementación pueden ser hechos a la librería sin romper la compatibilidad binaria. Este patrón tiene como objetivo conservar el tamaño de todas las clases públicas de las librerías constantes solo con agregar un único puntero. Este puntero apunta a la estructura de datos privada/interna que contiene todos los datos. El tamaño de esta estructura interna puede disminuir o aumentar sin tener ningún efecto adverso en la aplicación porque el puntero es accedido solo en el código de la librería y desde el punto de vista de la aplicación el tamaño del objeto nunca cambia, es siempre el tamaño del puntero (Qt Project, 2012). Este patrón se puede evidenciar en la figura 9 donde la clase *ImportWidget* posee un atributo de la clase *ImportWidgetPrivate* y crea una instancia de la misma.

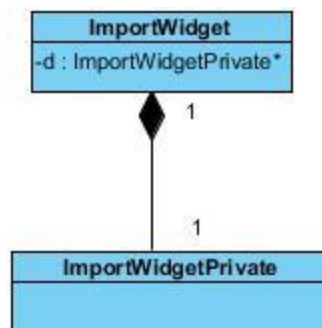


Fig. 9.-Patrón D-pointer.

2.5. Conclusiones

Al concluir el desarrollo de este capítulo se obtuvo como resultado una mejor definición del modelo conceptual sobre el que se basa el presente trabajo de diploma, contribuyendo a un mejor dominio del problema por parte del equipo de desarrollo. Los artefactos obtenidos en este capítulo facilitan que en el futuro la herramienta pueda ser extendida y perfeccionada. La utilización de patrones de diseño en la implementación de la solución propicia un ahorro de tiempo en el desarrollo. Con el uso de la metodología *XP* se construye un software con un diseño simple pero adecuado, lo que permite al sistema prepararse para cualquier situación.

CAPÍTULO III: Implementación y evaluación de la herramienta para la representación de los perfiles de los pozos de perforación minera

La fase de implementación en XP tiene su punto de partida en las tareas de ingeniería, a partir de estas los programadores implementan las historias de usuario. En este capítulo se definen las tareas de ingeniería, un conjunto de estándares de codificación para lograr una mejor estructuración del código y se muestran algunas de las interfaces obtenidas una vez implementada la aplicación.

En *XP* se definen un conjunto de normas para la validación de los productos, de manera que se pueda verificar su correcto funcionamiento. La validación del sistema se hace con el objetivo de asegurar que todos los requisitos previamente seleccionados están correctamente implementados, que se detecten las inconsistencias, omisiones, errores y que estos sean corregidos. En el capítulo se definen los casos de prueba, se les aplican a la herramienta, se corrigen los errores encontrados y se analizan los resultados obtenidos.

3.1. Implementación de la aplicación

La implementación en *XP* tiene varias características particulares de la propia metodología. Las tareas de ingeniería generadas por las historias de usuario se desarrollan siguiendo las siguientes prácticas:

- Adoptar un método de desarrollo basado en pruebas para asegurar que el código se comporta según lo esperado.
- Programación por parejas, para incrementar el conocimiento, la experiencia y las ideas.
- Asumir la propiedad colectiva del código, para que todo el equipo sea responsable de él.
- Integración continua, para reducir el impacto de la incorporación de nuevas funcionalidades.

3.1.1. *Tareas de ingeniería*

Una vez que se tienen las historias de usuario existe una herramienta que permite asignar las tareas relacionadas con cada historia de usuario a los involucrados del proyecto. Estas asignaciones se hacen mediante las tareas de ingeniería, que no son más que la representación gráfica de las responsabilidades asignadas, de cada miembro del equipo de desarrollo en *XP*.

Las tareas de ingeniería son consideradas como el punto de partida para los programadores. Cada tarea de ingeniería especifica el programador responsable, fecha de inicio y fin, el tiempo estimado en semana, una descripción de la misma y se especifica la historia de usuario que la generó. A continuación se muestran algunas de las tareas de ingeniería asociadas a las historias de usuario descritas anteriormente, en los artefactos generados se pueden ver las restantes tareas de ingeniería generadas en la construcción de la herramienta:

Tabla 8. Tarea de ingeniería #1.

| Tarea de Ingeniería | |
|---|--------------------------------------|
| Número Tarea: 1 | Número Historia de Usuario: 1 |
| Nombre Tarea: Crear interfaz para conectar aplicación con la base de datos. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 0.1 |
| Fecha Inicio: 01/12/2013 | Fecha Fin: 02/12/2013 |
| Programador Responsable: Yoan Mandina Verdecia | |
| Descripción: Se desarrolla la interfaz para que el usuario pueda conectar la aplicación con algunas de las bases de datos existentes. El usuario debe propiciar el servidor, puerto, usuario y contraseña. | |

Tabla 9. Tarea de ingeniería #2.

| Tarea de Ingeniería | |
|---|--------------------------------------|
| Número Tarea: 2 | Número Historia de Usuario: 1 |
| Nombre Tarea: Implementar funcionalidad para conectar aplicación con la base de datos. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 0.3 |
| Fecha Inicio: 02/12/2013 | Fecha Fin: 04/12/2013 |

| |
|---|
| Programador Responsable: Yoan Mandina Verdecia |
| Descripción: Al hacer clic en la opción Conectar del menú principal se debe mostrar la interfaz para conectar con la base de datos. Se debe implementar la funcionalidad para conectar la aplicación con la base de datos. Una vez que se establezca la conexión se muestran las bases de datos existentes y el usuario selecciona una y la aplicación carga los datos de dicha base de datos. |

Tabla 10. Tarea de ingeniería #3.

| Tarea de Ingeniería | |
|--|--------------------------------------|
| Número Tarea: 3 | Número Historia de Usuario: 2 |
| Nombre Tarea: Extraer información de la base de datos. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 0.2 |
| Fecha Inicio: 04/12/2013 | Fecha Fin: 06/12/2013 |
| Programador Responsable: Yoan Mandina Verdecia | |
| Descripción: Se ejecutan las consultas necesarias para obtener la información geológica minera de todos los pozos y se almacenan en estructuras de datos para su posterior uso. | |

Tabla 11. Tarea de ingeniería #4.

| Tarea de Ingeniería | |
|---|--------------------------------------|
| Número Tarea: 4 | Número Historia de Usuario: 2 |
| Nombre Tarea: Representar la columna litológica. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1.5 |
| Fecha Inicio: 06/12/2013 | Fecha Fin: 20/12/2013 |

Programador Responsable: Yoan Mandina Verdecia

Descripción: A partir de la información geológica minera de un pozo se representa la columna litológica correspondiente. Inicialmente se debe seleccionar un pozo de forma aleatoria.

3.1.2. Estándares de codificación

Usar técnicas de codificación y realizar buenas prácticas de programación con vistas a generar un código legible y de alta calidad es de gran importancia para la calidad del software y para mejorar el rendimiento. La aplicación de forma continuada de un estándar de codificación bien definido y la utilización de técnicas de programación apropiadas permiten que una aplicación informática se convierta en un sistema fácil de comprender y de mantener.

Con el objetivo de facilitar la lectura, comprensión, mantenimiento del código, estructura y apariencia física se utilizan estándares de codificación en la programación de la aplicación. Una clase tendrá la siguiente estructura:

```
class UnaClase
{
//ENUMERATIVES
    enum Enumerativo {Tipo1, Tipo2};

//METHODS
public:
    UnaClase();
    ~UnaClase();
    int attribute();
    void setAttribute(int value);

    int operation1(int arg1, int arg2);

private:
    int operation2();

//SLOTS
public slots:
    void onCreate();

//ATTRIBUTES
private:
    int _attribute;

//SIGNALS
signals:
    void sgCreated();
};
```

Fig. 10.-Estructura de una clase.

Identificadores

Los identificadores son los nombres de variables (arreglos, apuntadores), funciones, estructuras, clases, entre otras. Dichos identificadores siguen las siguientes normas, además de las definidas por el propio lenguaje.

- Deberán tener un nombre significativo para que por su simple lectura pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándares para que éstos tengan una longitud razonable. Si usa abreviaturas se debe manejar la misma lógica en todo el programa.
- Se utilizará la notación *Camel*, específicamente se utilizará la variante *lowerCamelCase* donde la primera letra se pone en minúscula.

3.1.3. Interfaces de la aplicación.

A continuación se muestran algunas interfaces de la aplicación consideradas las más importantes. En la figura 11 se muestra la interfaz para que el usuario pueda conectar la base de datos con la aplicación. En esta interfaz el usuario inserta los datos necesarios para la conexión.

La figura 12 muestra la interfaz para importar un archivo determinado, en esta se puede apreciar cómo los datos que se van a importar son mostrados al usuario haciendo uso de una tabla, de manera que el usuario pueda enlazar los datos que se desean importar con los que se esperan en la base de datos.

La figura 13 muestra la interfaz donde se representa la columna litológica y las concentraciones de los distintos minerales que posee un pozo determinado. En la parte izquierda se pueden apreciar los distintos pozos que contiene la base de datos y en la parte derecha se muestran los componentes y sus propiedades para que el usuario tenga la posibilidad de gestionar estos componentes.

La figura 14 muestra la interfaz donde se representa la estructura litológica del terreno, lo cual es una representación de las columnas litológicas de todos los pozos de perforación que contiene la base de datos.

La figura 15 muestra la interfaz donde se representa la localización de los pozos en el terreno, lo cual es una representación de todos los pozos que contiene la base de datos de acuerdo a sus coordenadas.

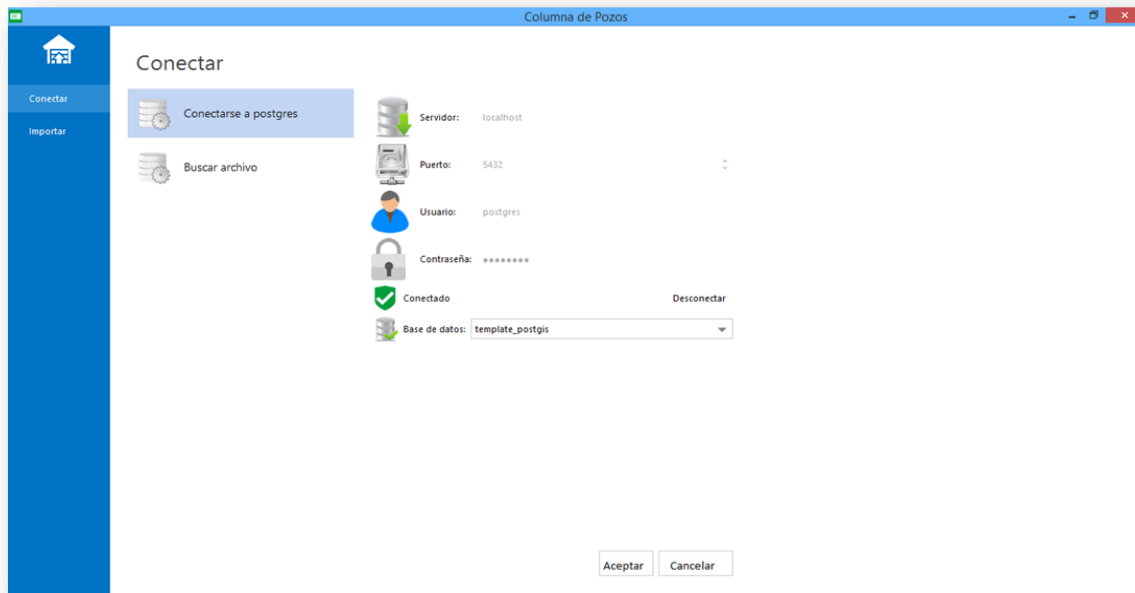


Fig. 11.-Interfaz para conectar aplicación con la base de datos.

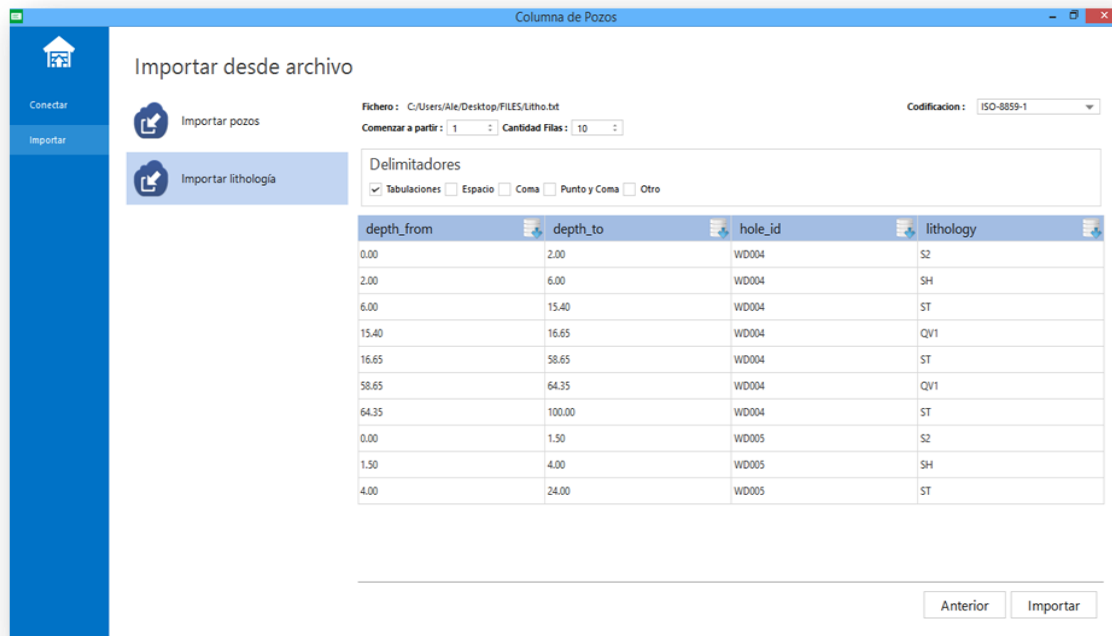


Fig. 12.-Interfaz que muestra cómo se importan los datos desde un archivo.

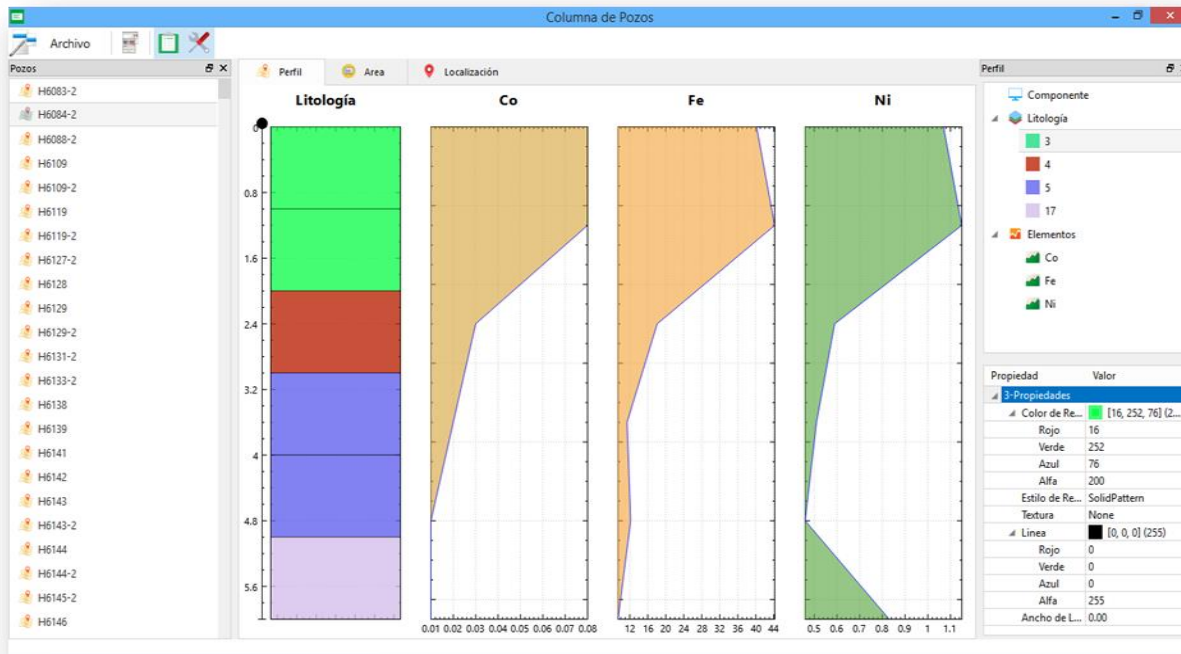


Fig. 13.-Interfaz que muestra el perfil litológico de un pozo determinado.

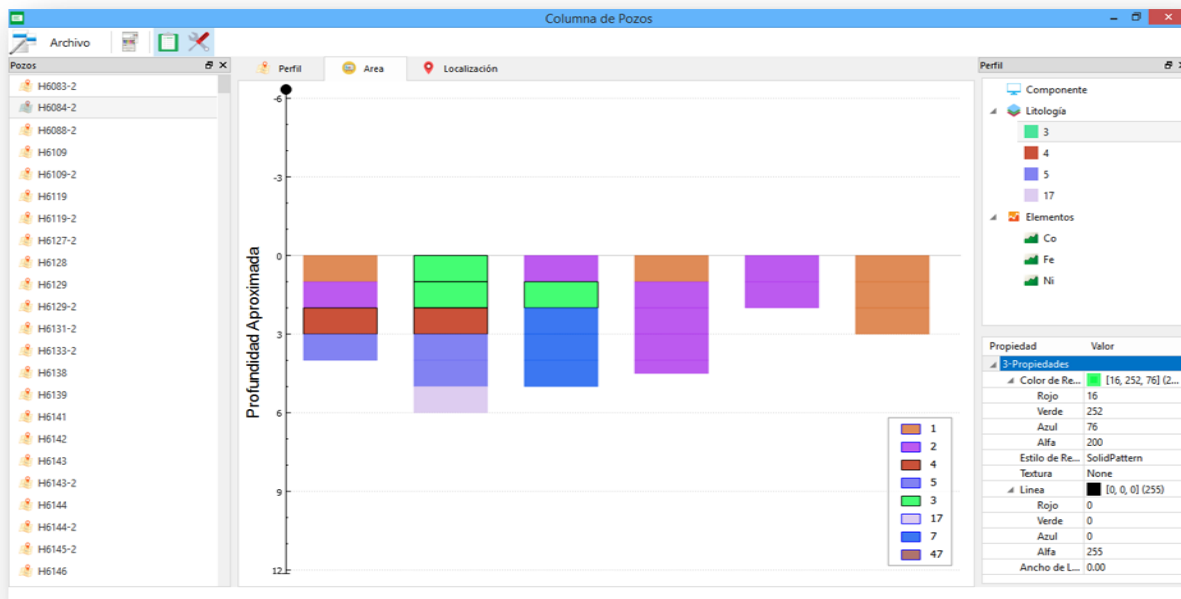


Fig. 14.-Interfaz que muestra la estructura litológica del terreno.

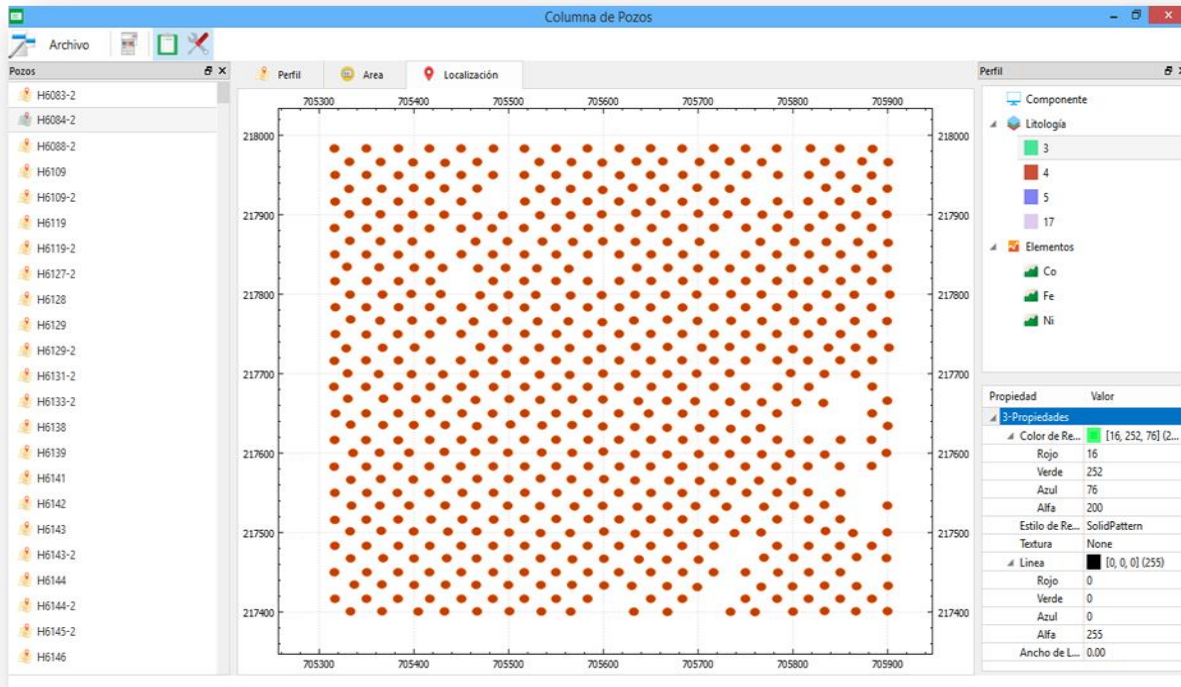


Fig. 15.-Interfaz que muestra la localización de los pozos en el terreno.

3.2. Proceso de pruebas de la metodología XP

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requisitos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. Es por ello que las pruebas de software son un elemento crítico para garantizar la calidad del software y representan una revisión final de las especificaciones del diseño.

3.2.1. Tipos de pruebas

Existen disímiles tipos de pruebas, este trabajo se ha centrado en la investigación de las pruebas correspondientes a la metodología de desarrollo XP, a continuación veremos algunas de estas.

Pruebas unitarias

Una prueba unitaria es la verificación de un módulo determinado dentro de un sistema. Son llevadas a cabo por los programadores encargados de cada módulo. Aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema.

De las pruebas unitarias se puede resumir:

- Brindan al desarrollador una inmediata retroalimentación de cómo está realizando su trabajo.
- El desarrollador puede realizar cambios de forma segura respaldada por efectivos casos de prueba.

Casos de prueba

Estas pruebas son definidas para cada historia de usuario y tienen el objetivo de asegurar que las funcionalidades del sistema cumplen con lo esperado. Estas pruebas no se realizan durante el desarrollo, sino que se realizan al producto terminado o a una iteración de este.

Es casi imposible obtener una implementación libre de errores y no conformidades, por lo tanto se debe tener definido un criterio de aprobación que indique cuando el producto cumple con lo que se espera de este. Después de realizado un estudio de algunas de las pruebas de software que propone la metodología *XP* se decide aplicar los casos de prueba con el propósito de indicarle al equipo de desarrollo cuando las funcionalidades implementadas en una iteración han sido completadas correctamente.

3.2.2. Técnicas de pruebas

Después de definidas las pruebas que se le van a aplicar al software es necesario contar con métodos y técnicas para aplicar cada una de estas pruebas. Son diversos los métodos y técnicas utilizadas para aplicar las pruebas, son comunes las técnicas de pruebas de caja blanca y pruebas de caja negra.

Pruebas de caja blanca

Las pruebas estructurales o de caja blanca es una técnica de diseño con casos de pruebas que permiten comprobar el código de la aplicación. Consiste en desarrollar pruebas que aseguren que las operaciones internas están acorde a las especificaciones establecidas y que cada componente interno se ha probado adecuadamente de manera individual.

En las pruebas estructurales las pruebas se seleccionan en función del conocimiento que se tiene de la implementación del módulo. Se suelen aplicar a módulos pequeños. El probador analiza el código y

deduce cuántos y qué conjuntos de valores de entrada han de probarse para que al menos se ejecute una vez cada sentencia del código (Drake, y otros, 2009).

Pruebas de caja negra

Las pruebas de funcionalidad, pruebas de comportamiento o pruebas de caja negra se basan en la especificación del programa a ser probado para elaborar los casos de prueba. El nombre de caja negra proviene de que el comportamiento del software solo puede ser determinado estudiando sus entradas y salidas a partir de casos de prueba (Juristo, y otros, 2006).

La figura 16 y figura 17 muestran algunos de los tipos de prueba que usualmente se aplican usando las técnicas de pruebas de caja blanca y pruebas de caja negra respectivamente:



Fig. 16.-Tipos de pruebas de caja blanca.



Fig. 17.-Tipos de pruebas de caja negra.

En la herramienta para la representación de los perfiles de los pozos de perforación minera, se decidió aplicar la técnica de pruebas de caja negra y dentro de estas las pruebas funcionales, ya que este tipo de prueba permite comprobar la correcta implementación de las funcionalidades.

3.3. Descripción del proceso de pruebas

Con los casos de prueba se puede verificar si la aplicación realmente funciona. En estos se especifica la historia de usuario a la que se le va a aplicar, nombre de la persona que realiza la prueba, condiciones

necesarias para que pueda ejecutarse la prueba, los pasos que deben seguirse en su ejecución, el resultado que se espera de la aplicación y una evaluación final por el que aplica la prueba.

A continuación se muestran algunos de los casos de pruebas aplicados al software, en los artefactos generados se muestran los restantes casos de pruebas.

Tabla 12. Caso de prueba CP1_HU1.

| Caso de Prueba | | |
|---|--|---------------------|
| Código de Caso de Pruebas: CP1_HU1 | Número de la Historia de Usuario: 1 | Iteración: 1 |
| Nombre de Caso de Prueba: Comprobar que la aplicación se conecta a la base de datos. | | |
| Nombre de la persona que realiza la prueba: Carlos Alberto Otero Alguacil. | | |
| Descripción de la prueba: Esta prueba consiste en comprobar que la aplicación se conecta correctamente a una base de datos determinada. | | |
| Condiciones de ejecución: Para poder realizar la prueba es necesario que se encuentren instalados el sistema gestor de bases de datos <i>PostgreSQL</i> y el módulo <i>PostGIS</i> . También debe existir al menos una base de datos creada. | | |
| <p>Entrada/Pasos de ejecución: Se debe acceder al menú <u>Archivo</u> y seleccionar la opción <u>Conectar</u>, luego se debe seleccionar la opción desplegada <u>Conectar a base de datos</u>. El usuario debe proporcionar los siguientes datos:</p> <ul style="list-style-type: none"> - Servidor - Puerto - Usuario - Contraseña <p>Una vez proporcionado estos datos el usuario debe dar clic en el botón <u>Conectar</u>, si los datos proporcionados son correctos la aplicación debe mostrar las bases de datos existentes y mostrar el estado CONECTADO, en caso contrario debe mostrar el estado DESCONECTADO. El usuario también puede dar clic en el botón <u>Desconectar</u> y el sistema debe mostrar el estado</p> | | |

| |
|---|
| <p>DESCONECTADO.</p> <p>Posteriormente el usuario debe seleccionar una de las bases de datos y dar clic en el botón <u>Ok</u>.</p> <p>En caso de que el usuario no quiera conectar la aplicación a la base de datos debe hacer clic en el botón <u>Cancelar</u> y la ventana debe cerrarse automáticamente.</p> |
| <p>Resultado Esperado: Automáticamente debe mostrarse en la ventana principal el módulo de visualización, que contiene la columna litológica y la concentración de cada uno de los minerales que posee uno de los pozos de la base de datos seleccionada.</p> |
| <p>Evaluación de la prueba: Prueba satisfactoria.</p> |

Tabla 13. Caso de prueba CP1_HU2.

| Caso de Prueba | | |
|--|--|--------------------|
| Código de Caso de Pruebas : CP1_HU2 | Número de la Historia de Usuario: 2 | Iteración:1 |
| Nombre de Caso de Prueba: Comprobar que la columna litológica se representa correctamente. | | |
| Nombre de la persona que realiza la prueba: Carlos Alberto Otero Alguacil. | | |
| Descripción de la prueba: Esta prueba consiste en comprobar que la columna litológica de un pozo determinado se muestra correctamente. | | |
| Condiciones de ejecución: La aplicación debe estar conectada a alguna base de datos. | | |
| Entrada/Pasos de ejecución: En la parte derecha de la interfaz debe aparecer una lista de todos los pozos existentes de la base de datos a la que está conectada la aplicación. Para ver la columna litológica de un pozo el usuario debe seleccionar uno de los pozos y la columna litológica debe representarse en la parte derecha de la interfaz principal. | | |

| |
|---|
| <p>Resultado Esperado: La columna litológica debe estar representada en una gráfica de coordenada enumerada con los valores invertidos de manera que represente la profundidad del pozo. La columna litológica tiene varias litologías y cada una de estas tiene un color determinado, de manera que litologías similares poseen el mismo color.</p> |
| <p>Evaluación de la prueba: Prueba satisfactoria.</p> |

Tabla 14. Caso de prueba CP1_HU3.

| Caso de Prueba | | |
|--|--|--------------------|
| Código de Caso de Pruebas : | Número de la Historia de Usuario: 3 | Iteración:1 |
| CP1_HU3 | | |
| <p>Nombre de Caso de Prueba: Comprobar que la concentración de cada uno de los minerales se representa correctamente.</p> | | |
| <p>Nombre de la persona que realiza la prueba: Carlos Alberto Otero Alguacil.</p> | | |
| <p>Descripción de la prueba: Esta prueba consiste en comprobar que la concentración de cada uno de los minerales de un pozo determinado se muestra correctamente.</p> | | |
| <p>Condiciones de ejecución: La aplicación debe estar conectada a alguna base de datos.</p> | | |
| <p>Entrada/Pasos de ejecución: En la parte derecha de la interfaz debe aparecer una lista de todos los pozos existentes de la base de datos a la que está conectada la aplicación. Para ver la concentración de cada uno de los minerales que posee un pozo determinado el usuario debe seleccionar uno de los pozos y las concentraciones de cada uno de los minerales deben representarse a la derecha de la columna litológica del pozo.</p> | | |
| <p>Resultado Esperado: Debe aparecer una gráfica de línea que represente la concentración por cada uno de los minerales que posea dicho pozo. Los intervalos del eje de coordenada de la gráfica de la columna litológica deben coincidir con los intervalos del eje de coordenada de</p> | | |

la gráfica de las concentraciones de cada uno de los minerales. Cada gráfica de concentración de un mineral debe poseer un color de fondo y un color de borde de gráfica asignado. En la parte posterior de la gráfica debe aparecer el nombre del mineral que está siendo representada la concentración. En la parte inferior de la gráfica debe aparecer el eje X enumerado de manera que pueda apreciarse el valor de la concentración en un intervalo determinado.

Evaluación de la prueba: Prueba satisfactoria.

3.4. Análisis de los resultados del proceso de prueba

Los casos de pruebas fueron aplicados a las 18 historias de usuarios implementadas permitiendo detectar algunos errores cometidos en la implementación del sistema. Al finalizar cada iteración fueron aplicados los casos de pruebas correspondientes a las historias de usuarios implementadas en esa iteración y se repitieron las pruebas en las historias de usuarios que se encontraron no conformidades, para comprobar que fueron arregladas correctamente. La siguiente tabla muestra un resumen de los casos de pruebas aplicados en cada iteración, las no conformidades encontradas y las no conformidades resueltas:

Tabla 15. Resultados de los casos de pruebas aplicados.

| Resultados Obtenidos | | | |
|----------------------|---|------------------------------|----------------------------|
| Iteración | Casos de Pruebas | No conformidades encontradas | No conformidades resueltas |
| 1 | CP1_HU1, CP1_HU2, CP1_HU3, CP1_HU4, CP1_HU7, CP1_HU10 | 3 | 3 |
| 2 | CP1_HU5, CP1_HU6, CP2_HU7, CP1_HU8, CP1_HU9, CP2_HU10, CP1_HU11, CP1_HU12, CP1_HU13 | 5 | 5 |
| 3 | CP_HU5, CP_HU6, CP_HU8, CP_HU9, CP_HU10, CP_HU11, CP_HU12, CP_HU13, CP_HU14, CP_HU15, CP_HU16, CP_HU17, | 2 | 2 |

| | | | |
|--|---------|--|--|
| | CP_HU18 | | |
|--|---------|--|--|

3.5. Conclusiones

Los estándares de codificación utilizados en la implementación de la herramienta son indispensables y posibilitan un mejor entendimiento y organización del código. Una vez terminada la herramienta se puede apreciar que las interfaces de la aplicación tienen un diseño que posibilita una fácil interacción con el usuario, además la herramienta construida tiene competitividad respecto a las otras herramientas analizadas ya que tiene en cuenta la representación de la estructura litológica del terreno y la localización de los pozos en el terreno. Por otro lado los casos de pruebas son adecuados, pues garantizan que la aplicación funcione correctamente teniendo en cuenta lo que se espera de esta.

Conclusiones Generales

Como conclusiones del presente trabajo de investigación se tiene:

- Las soluciones existentes analizadas no responden totalmente al problema de la investigación ya que son propietarias y suponen un problema para las empresas que las utilizan.
- Las tecnologías, herramientas, lenguajes y *framework* de desarrollo seleccionados para el desarrollo de la aplicación obedecen a criterios de selección de tecnologías libres garantizando las políticas que impulsa la universidad y el país.
- La documentación técnica generada durante la investigación facilita que un futuro la herramienta pueda ser ampliada y perfeccionada.
- Se cumplió el objetivo planteado logrando desarrollar una herramienta informática que permite la representación de los perfiles de los pozos de perforación minera para el estudio y análisis de los yacimientos minerales.
- La herramienta construida tiene competitividad respecto a las otras herramientas analizadas ya que tiene en cuenta la representación de la estructura litológica del terreno y la localización de los pozos en el terreno.
- Los casos de pruebas son adecuados, pues garantizan que la aplicación funcione correctamente teniendo en cuenta lo que se espera de esta.

Recomendaciones

Los autores recomiendan:

Implementar el proceso de análisis en la herramienta para la representación de los perfiles de los pozos de perforación minera.

Referencias Bibliográficas

Beck, Kent y Fowler, Martin. 2000. *Planning Extreme Programming*. Boston : Addison-Wesley Longman, 2000. ISBN:0201710919 .

Chávez Márquez, Nilberto Caridad, Ramírez Cruz, Eduardo y Ramírez, Dainy. 2012. Serie Científica. *Análisis del sistema para graficar columnas litológicas de pozos de petróleo*. [En línea] 2012. [Citado el: 10 de Diciembre de 2013.] <http://publicaciones.uci.cu/index.php/SC>. ISSN:2306-2495.

Dabrio González, Cristino José and Hernando, Santiago. 2003. *Estratigrafía*. Madrid : Universidad Complutense, 2003. ISBN: 84-600-9887-7.

Drake, José M. y López, Patricia. 2009. *Verificación y Validación*. España : s.n., 2009.

Eichhammer, Emanuel. 2006. QCustomPlot. [En línea] 2006. [Citado el: 2 de Enero de 2014.] <http://www.qcustomplot.com>.

Environmental Systems Research Institute. 2004. Rock Ware. [En línea] 2004. [Citado el: 20 de marzo de 2014.] <https://www.rockware.com/product>.

Gallo, Elisa y Vergara, Mikel. 2012. European Software Institute. *Metodologías de Desarrollo de Software Ágiles*. [En línea] 14 de Junio de 2012. [Citado el: 28 de Enero de 2014.] <http://www.esi.es/>.

Hass Mora, Hugo Sergio. 2010. Facultad de Ingeniería. *Exploración y muestreo de los geomateriales*. [En línea] 2010. [Citado el: 12 de Enero de 2014.] <http://www.ingenieria.unam.mx/>.

Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2006. Grupo de Investigación en Ingeniería del Software Empírica . *Técnicas de evaluación de software*. [En línea] 17 de octubre de 2006. [Citado el: 22 de Enero de 2014.] <http://www.grise.upm.es/>.

Kendall, Julie E. y Kendall, Kenneth E. 2005. *Análisis y Diseño de Sistemas*. México : PEARSON EDUCACION, 2005. ISBN:970-26-0577-6.

REFERENCIAS BIBLIOGRÁFICAS

Letelier, Patricio y Penadés, M. Carmen. 2010. Facultad de Informática. *Métodologías ágiles para el desarrollo de software*. [En línea] Septiembre de 2010. [Citado el: 26 de Noviembre de 2013.] www.willydev.net.

Lorente, César de la Torre, y otros. 2010. *Guía de Arquitectura N-Capas orientada al dominio con .NET*. España : Krasis Press, 2010. ISBN: 978-84-936696-3-8

Momjian, Bruce. 2000. *PostgreSQL: Introduction and Concepts*. United States of America : Addison Wesley, 2000. ISBN: 0201703319.

Pressman, Roger. 2002. *Ingeniería del Software, un enfoque práctico*. Madrid: McGRAW-HILL S.A.U., 2002. ISBN:84-481-3214-9.

Qt Project. 2012. Qt Project. *D-Pointer*. [En línea] 2012. [Citado el: 5 de Febrero de 2014.] <http://qt-project.org/>.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000. *El Lenguaje Unificado de Modelado*. Madrid : En español-Pearson Educación S.A, 2000. ISBN:8478290370.

Solís, Manuel Calero. 2003. *Una explicación de la programación extrema (XP)*. Madrid: s.n., 2003.

Suárez, Pablo y Fonseca, Carlos. 2003. Facultad de Ingeniería. *Documentación y pruebas ante el paradigma de objetos*. [En línea] 2003. [Citado el: 3 de Febrero de 2014.] <http://campus.fi.uba.ar/>.

Tercera convención cubana de las ciencias de la tierra. Geociencias. 2009. La Habana : s.n., 2009. ISBN: 978-959-71117-19-3.

Theelin, Johan. 2007. *Foundations of Qt Development*. New York: Springer Verlag GmbH, 2007. ISBN:978-1-59059-831-3.

Tomás Jover, Roberto, et al. 2004. *Diseño de un modelo geológico-geotécnico 3D de la Vega Baja del Río Segura*. Valencia : Universidad de Zaragoza, 2004. ISBN:84-95475-39-1.

Valderrey Sanz, Pablo. 2011. *Administración de Sistemas Gestores de Base de Datos*. Madrid : STARBOOK EDITORIAL, 2011. ISBN: 9788492650781.

Weller, J. M. 1960. *Stratigraphic Principles and Practice*. New York: Harper & Brothers, 1960. HBE27-59.

Bibliografía Consultada

Beck, Kent and Fowler, Martin. 2000. *Planning Extreme Programming*. Boston: Addison-Wesley Longman, 2000. ISBN:0201710919 .

Berson, A. 1996. *Client/Server Architecture*. s.l. : McGraw-Hill, 1996. ISBN: 0070056641.

Bolaños Alonso, Daniel, Sierra Alonso, Almudena and Alarcón Rodríguez, Miren Idoia. 2007. *Pruebas de software y junit*. Madrid: Pearson Prentice Hall, 2007. ISBN: 8483223546.

Castillo Sánchez, Mauricio. 2000. *Manual para la formación de investigadores : una guía hacia el desarrollo del espíritu científico*. Bogotá: Cooperativa Editorial Magisterio, 2000. ISBN: 958-20-0499-1.

Chávez Márquez, Nilberto Caridad, Ramírez Cruz, Eduardo and Ramírez, Dainy. 2012. Serie Científica. *Análisis del sistema para graficar columnas litológicas de pozos de petróleo*. [Online] 2012. [Cited: Diciembre 10, 2013.] <http://publicaciones.uci.cu/index.php/SC>. ISSN:2306-2495.

Cortada de Kohan, Nuria, Macbeth, Guillermo and López Alonso, Alfredo. 2008. *TÉCNICAS DE INVESTIGACIÓN CIENTÍFICA*. Buenos Aires: Lugar Editorial., 2008. ISBN: 9789508922977.

Dabrio González, Cristino José and Hernando, Santiago. 2003. *Estratigrafía*. Madrid: Universidad Complutense, 2003. ISBN: 84-600-9887-7.

de la Cuadra Irizar, Luis. 1974. *Laboreo de Minas*. Madrid : AUTOR-EDITOR, 1974. ISBN:9788460001201.

Drake, José M. and López, Patricia. 2009. *Verificación y Validación*. España : s.n., 2009.

Eichhammer, Emanuel. 2006. QCustomPlot. [En línea] 2006. [Citado el: 2 de Enero de 2014.] <http://www.qcustomplot.com>.

Environmental Systems Research Institute. 2004. Rock Ware. [En línea] 2004. [Citado el: 20 de Marzo de 2014.] <https://www.rockware.com/product>.

BIBLIOGRAFÍA CONSULTADA

- Gallo, Elisa and Vergara, Mikel. 2012.** European Software Institute. *Metodologías de Desarrollo de Software Ágiles*. [Online] Junio 14, 2012. [Cited: Enero 28, 2014.] <http://www.esi.es/>.
- Gómez Fuentes, María del Carmen. 2011.** *Notas del Curso Análisis de Requerimientos*. México : Universidad Autonoma Metropolitana, 2011. ISBN:978-607-477-442-9.
- Hass Mora, Hugo Sergio. 2010.** Facultad de Ingeniería. *Exploración y muestreo de los geomateriales*. [Online] 2010. [Cited: Enero 12, 2014.] <http://www.ingenieria.unam.mx/>.
- Hernández Sampieri, Roberto, Fernández Collado, Carlos and Baptista Lucio, Pilar. 2006.** *Metodología de la Investigación*. Abastos : McGRAW-HILL, 2006. ISBN:970-10-5753-8.
- Humphrey, Watts S. 1995.** *A Discipline for software engineering*. United States of America : Addison-Wesley, 1995. ISBN: 0-201-54610-8.
- Juristo, Natalia, Moreno, Ana M. and Vegas, Sira. 2006.** Grupo de Investigación en Ingeniería del Software Empírica . *Técnicas de evaluación de software*. [Online] octubre 17, 2006. [Cited: Enero 22, 2014.] <http://www.grise.upm.es/>.
- Kendall, Julie E. and Kendall, Kenneth E. 2005.** *Análisis y Diseño de Sistemas*. México : PEARSON EDUCACION, 2005. ISBN:970-26-0577-6.
- Kniberg, Henrik. 2007.** *Scrum y XP desde las trincheras*. Estados Unidos de América. : C4Media., 2007. ISBN: 978-1-4303-2264-1.
- Larman, Craig. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999. ISBN:970-17-0261-1.
- Letelier, Patricio and Penadés, M. Carmen. 2010.** Facultad de Informática. *Métodologías ágiles para el desarrollo de software*. [Online] Septiembre 2010. [Cited: Noviembre 26, 2013.] www.willydev.net.
- Lorente, César de la Torre, et al. 2010.** *Guía de Arquitectura N-Capas orientada al dominio con .NET*. España : Krasis Press, 2010. ISBN: 978-84-936696-3-8.

BIBLIOGRAFÍA CONSULTADA

Momjian, Bruce. 2000. *PostgreSQL: Introduction and Concepts*. United States of America : Addison Wesley, 2000. ISBN: 0201703319.

Oualine, Steve. 2002. *Practical C++ Programming*. s.l. : O'Reilly, 2002. 0-596-00419-2.

Pressman, Roger. 2002. *Ingeniería del Software, un enfoque práctico*. Madrid : McGRAW-HILL S.A.U., 2002. ISBN:84-481-3214-9.

—. **2010.** *Ingeniería del software. Un enfoque práctico. Séptima edición*. México : McGRAW-HILL, 2010. ISBN: 978-607-15-0314-5.

Qt Project. 2012. Qt Project. *D-Pointer*. [Online] 2012. [Cited: Febrero 5, 2014.] <http://qt-project.org/>.

Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000. *El Lenguaje Unificado de Modelado*. Madrid : En español-Pearson Educación S.A, 2000. ISBN:8478290370.

Solís, Manuel Calero. 2003. *Una explicación de la programación extrema (XP)*. Madrid : s.n., 2003.

Suárez, Pablo and Fonseca, Carlos. 2003. Facultad de Ingeniería. *Documentación y pruebas ante el paradigma de objetos*. [Online] 2003. [Cited: Febrero 3, 2014.] <http://campus.fi.uba.ar/>.

Tercera convención cubana de las ciencias de la tierra. Geociencias. 2009. La Habana : s.n., 2009. ISBN: 978-959-71117-19-3.

Theelin, Johan. 2007. *Foundations of Qt Development*. New York : Springer Verlag Gmbh, 2007. ISBN:978-1-59059-831-3.

Tomás Jover, Roberto, et al. 2004. *Diseño de un modelo geológico-geotécnico 3D de la Vega Baja del Río Segura*. Valencia : Universidad de Zaragoza, 2004. ISBN:84-95475-39-1.

Valderrey Sanz, Pablo. 2011. *Administración de Sistemas Gestores de Base de Datos*. Madrid : STARBOOK EDITORIAL, 2011. ISBN: 9788492650781.

Weller, J. M. 1960. *Stratigraphic Principles and Practice*. New York : Harper & Brothers, 1960. HBE27-59.

Anexos

Anexo I: Historias de Usuario.

Tabla 16. Historia de Usuario #4.

| Historia de Usuario | |
|--|---|
| Número: 4 | Nombre de la Historia de Usuario: Mostrar los tipos de litología de un pozo y los nombres de los minerales existentes. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 1 |
| Prioridad en negocio: Muy Alta. | Puntos estimados(semanas): 0.5 |
| Riesgo en desarrollo: Media. | Puntos reales(semanas): 0.4 |
| Descripción: Se debe mostrar los elementos principales que posee el componente de visualización, estos son las barras litológicas que conforman la columna litológica y los nombres de los minerales a los cuales se le miden la concentración en dicho pozo. | |
| Observaciones: Se debe tener en cuenta que la finalidad de mostrar las barras que representan a las litologías y los minerales existentes es para poder personalizar posteriormente cada uno de estos de manera individual. | |

Tabla 17. Historia de Usuario #5.

| Historia de Usuario | |
|--|---|
| Número: 5 | Nombre de la Historia de Usuario: Editar propiedades de la columna litológica. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 2 |
| Prioridad en negocio: Alta. | Puntos estimados(semanas): 1.0 |
| Riesgo en desarrollo: Muy Alta. | Puntos reales(semanas): 1.0 |

Descripción: Una vez representada la columna litológica de un pozo, se debe permitir gestionar un conjunto de propiedades de dicha representación.

El sistema debe permitir:

- Editar el color de cada una de las barras litológicas.
- Aplicar una textura a cada una de las barras, para ello se debe proporcionar un listado de las texturas existentes que no son más que un conjunto de imágenes que caracterizan el tipo de roca perteneciente a dicha barra.

Observaciones: Se debe garantizar que una vez asignado un color a una barra o una textura, las barras del mismo tipo deben tomar las mismas propiedades.

Tabla 18. Historia de Usuario #6.

| Historia de Usuario | |
|--|--|
| Número: 6 | Nombre de la Historia de Usuario: Editar propiedades de las concentraciones de los minerales. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 2 |
| Prioridad en negocio: Alta. | Puntos estimados(semanas): 2.0 |
| Riesgo en desarrollo: Muy Alta. | Puntos reales(semanas): 2.5 |
| <p>Descripción: Una vez representada las concentraciones de cada uno de los minerales de un pozo, se debe permitir gestionar un conjunto de propiedades de dicha representación.</p> <p>El sistema debe permitir:</p> <ul style="list-style-type: none"> ➤ Editar el color de fondo de la gráfica que representa la concentración de cada uno de los minerales de manera independiente. ➤ Editar el color del borde de la gráfica que representa la concentración de cada uno de los minerales de manera independiente. ➤ Cambiar el estilo de gráfica de la representación de una concentración. Se debe brindar la posibilidad de representar la concentración en forma de barra y en forma de gráfica de línea como se muestra en el prototipo 1. ➤ Asignar estilo de dispersión a la gráfica de la representación de una concentración de estilo línea. | |

Esto significa que a cada punta de la gráfica se le debe asignar una figura como se muestra en el prototipo 2. Para esto se debe proporcionar un listado con los estilos de dispersión existentes.

Observaciones: Se debe tener en cuenta que los estilos de dispersión son solo para las gráficas de línea, es decir cuando el estilo de la gráfica de las concentraciones tenga el estilo de barra, esta opción no se debe permitir.

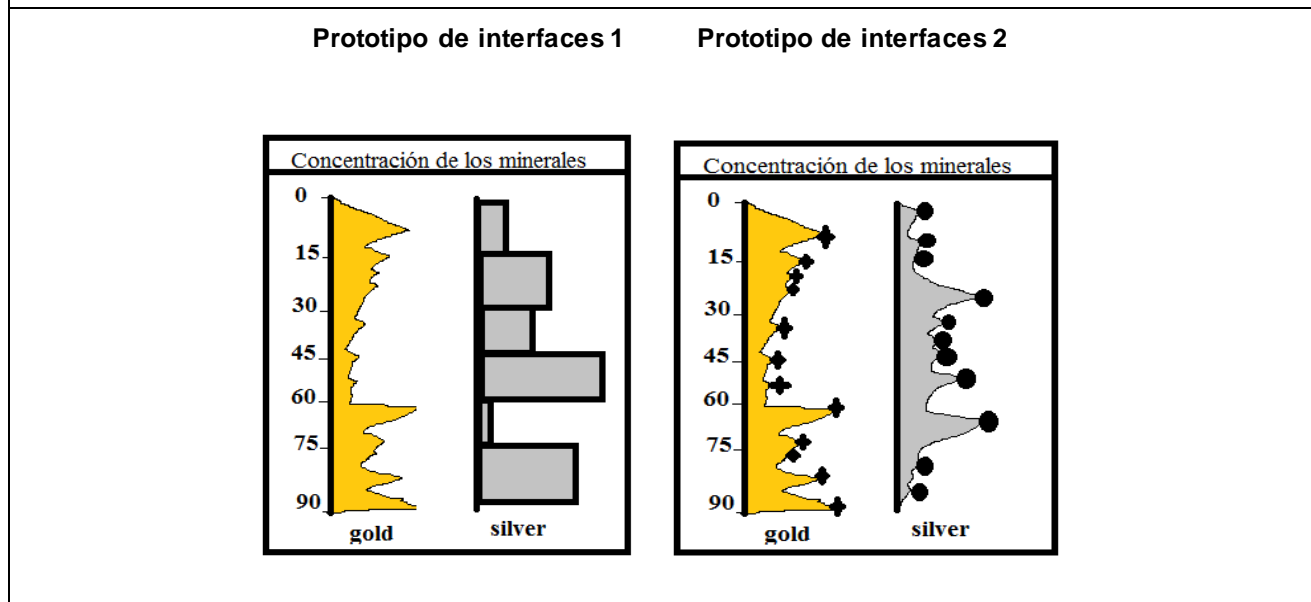


Tabla 19. Historia de Usuario #7.

| Historia de Usuario | |
|---|--|
| Número: 7 | Nombre de la Historia de Usuario: Representar estructura litológica del terreno. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 1 |
| Prioridad en negocio: Muy Alta. | Puntos estimados(semanas): 0.5 |
| Riesgo en desarrollo: Alta. | Puntos reales(semanas): 0.6 |
| <p>Descripción: Se debe brindar la posibilidad de representar la estructura litológica del terreno, lo cual significa representar la columna litológica de cada uno de los pozos existentes en la base de datos a la que</p> | |

está conectada la aplicación es ese momento, estos se mostrarán uno a continuación del otro, teniendo en cuenta que litologías similares tienen la misma configuración, la representación debe quedar como muestra el prototipo 1.

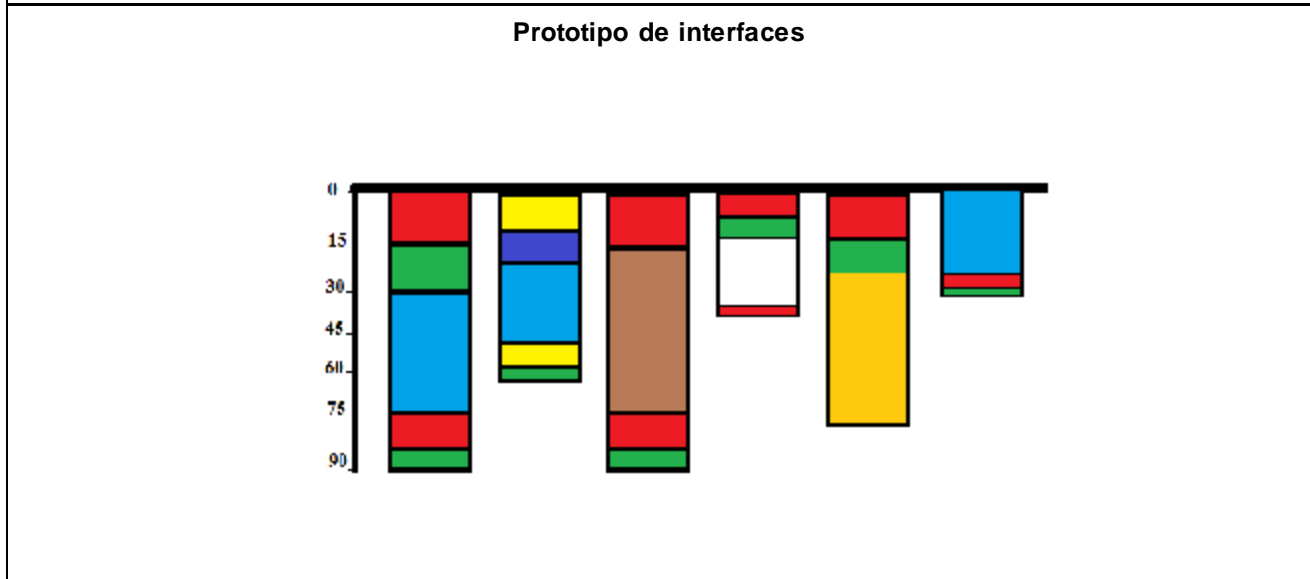


Tabla 20. Historia de Usuario #8.

| Historia de Usuario | |
|---|---|
| Número: 8 | Nombre de la Historia de Usuario: Exportar como PDF. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 2 |
| Prioridad en negocio: Alta. | Puntos estimados(semanas): 0.5 |
| Riesgo en desarrollo: Media. | Puntos reales(semanas): 0.3 |
| <p>Descripción: Una vez representada la columna litológica de un pozo y la concentración de cada uno de los minerales se debe permitir exportar esa información como archivo de extensión *.PDF, la información exportada debe verse similar a como describe el prototipo 1. También debe permitir exportar con esta extensión la estructura litológica del terreno como se aprecia en el prototipo 1 de la Historia de Usuario#7.</p> | |

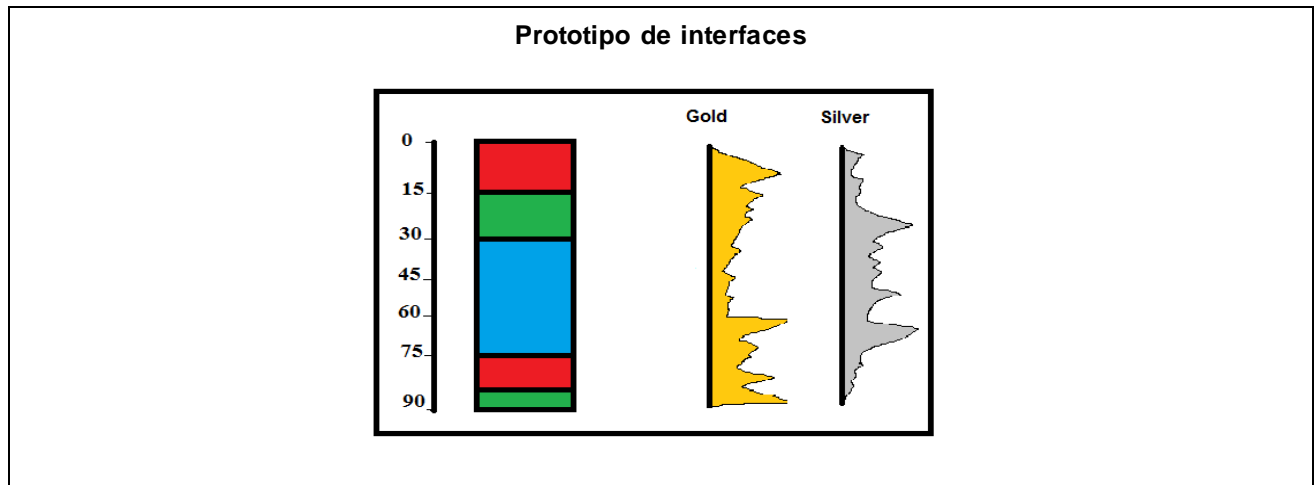


Tabla 21. Historia de Usuario #9.

| Historia de Usuario | |
|--|--|
| Número: 9 | Nombre de la Historia de Usuario: Mostrar longitud del pozo. |
| Cantidad de modificaciones a la Historia de Usuario: Ninguna. | |
| Usuario: Geólogo. | Iteración asignada: 2 |
| Prioridad en negocio: Alta. | Puntos estimados(semanas): 0.2 |
| Riesgo en desarrollo: Baja. | Puntos reales(semanas): 0.2 |
| Descripción: Debe mostrarse la longitud en profundidad del pozo que está representado. | |