

**Universidad de las Ciencias Informáticas**



**Facultad 5**

**Título: Implementación del Sistema para  
Graficar Columnas Litológicas de  
Pozos de Petróleo**

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

**Autor:** Adriel Linares Ramos

**Tutor:** Ing. David Tavares Cuevas.

Ciudad de la Habana

Junio del 2011

## **DECLARACIÓN DE AUTORÍA**

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Isabel González Flores**

\_\_\_\_\_

Firma del Autor

**Humberto Cardoza Rodríguez**

\_\_\_\_\_

Firma del Autor

## **DATOS DE CONTACTO**

### **Tutor**

**Nombre y apellidos:** David Tavares Cuevas

**Institución:** UCI

**Dirección de la institución:** Carretera de San Antonio. Boyeros. Ciudad Habana.

**Correo electrónico:** dtavares@uci.cu

**Teléfono del trabajo:** 8372331

**Categoría docente:** Adiestrado

**Cargo del trabajador:** Profesor. Líder de proyecto.

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas.

**Año de graduación:** 2009

**Institución donde se graduó:** UCI.

*Quisiera agradecer en general a todos los que de una forma u otra me han ayudado en la realización de este trabajo, en especial:*

*A mis padres por apoyarme en todos los momentos de mi vida, por ser la guía en tantos años de educación y porque sin ellos no hubiera logrado este sueño.*

*A mi hermana por su apoyo en tantos años de vida y por sus consejos que tanto me hicieron falta en estos 5 años.*

*A mi querida prima Danisa por acompañarme y ayudarme a superar momentos difíciles que pasamos juntos en años de estudio.*

*A mi familia que vive en la Habana por ayudarme en todo lo que necesité y al resto de mi familia que de una forma u otra me han ayudado en mis estudios.*

*A mis amigos del cuarto, Palomino, Andy, Siles y Tejeda por los buenos momentos que pasamos juntos y por las locuras realizadas en estos 5 años de Universidad.*

*A Lia por su gran paciencia y dedicación a la hora de corregirnos los errores de este trabajo.*

*A mis amigos de siempre Julio y Adrian por ser mis hermanos en estos 5 años y por todos los momentos de alegrías que compartimos.*

*A mi amiga Danae, por sus consejos, por estar presente en cada segundo que lo necesité y en especial por su maravillosa amistad.*

*A Liusba por ser esa mujer tan especial que todo hombre necesita tener en su vida, por sus consejos, su comprensión y por saber ser una amiga y a la vez una hermana para mí.*

*A Libeidy por su amistad y por todos esos momentos bonitos que compartimos que nunca olvidaré.*



*A Liudmila, Yoana, Irenna, Sandra y Dayanis por su amistad y por todos los momentos llenos de alegrías que compartimos.*

*A Kaliannis y Leiser que han sido esos amigos inolvidables que me acompañan desde el preuniversitario.*

*A todos, Gracias Siempre.*

*Este trabajo es uno de los grandes sueños de mi vida y de otras personas que han estado a mi lado siempre sin importar lo que suceda, es por ello que quiero dedicarle este trabajo:*

*A mis queridos padres porque sin ellos este sueño no se hubiese hecho realidad, porque este logro es tanto mío como de ellos, por apoyarme y darme las fuerzas necesarias para superar cada momento de esta vida.*

*A mi adorada hermana porque la quiero con la vida y porque desde niños ha sido esa luz que siempre ilumina mi camino.*



Este trabajo se realiza con el objetivo de materializar mejoras en la automatización del proceso de representación gráfica de columnas litológicas de pozos de petróleo. Para cumplir este, se implementa un sistema que permite representar la profundidad de los pozos de petróleo, los elementos que se representan a través de curvas, así como las Columnas Litológicas correspondientes a cada pozo. El sistema permite trabajar con varios proyectos en el área de trabajo, donde cada uno corresponde a un pozo de petróleo. Luego de terminado la construcción de las columnas se permite exportar el resultado en un documento PDF o en una imagen. Estos archivos contienen la información necesaria sobre las columnas litológicas de los pozos de petróleo, la cual es agregada al parte diario de geología. La realización de este sistema permite a los trabajadores del CEINPET realizar el proceso de representación gráfica de una forma sencilla y eficiente, ahorrándole a la empresa tiempo y dinero de inversión en otros software privativos e inapropiados.

**Palabras clave:**

Columnas, Curvas, Graficar, Litologías, Petróleo

**TABLA DE CONTENIDO**

INTRODUCCIÓN .....	1
CAPÍTULO 1: Fundamentación Teórica .....	6
1.1 Introducción .....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.3 Tipos de gráficos en perforación .....	8
1.4 Representación gráfica de Columnas Litológicas .....	11
1.4.1 Descripción general .....	11
1.4.2 Descripción actual del dominio del problema.....	13
1.5 Análisis de las soluciones existentes vinculadas al campo de acción.....	14
1.5.1 Descripción General.....	15
1.5.2 WellWizard .....	16
1.6.3 Wellsight System .....	16
1.5.4 WellFlo 3 .....	17
1.5.5 Strater .....	17
1.6 Estándares de codificación .....	19
1.7 Patrones.....	19
1.8 Estilos Arquitectónicos .....	20
1.9 Programación Orientada a Objetos (POO) .....	21
1.9.1 Características de la POO.....	21
1.9.2 Ventajas e inconvenientes de la POO.....	22
1.10 Conclusiones Parciales.....	23
Capítulo 2: Herramientas y tecnologías actuales a utilizar .....	24
2.1 Introducción .....	24
2.2 Tipo de Aplicación a construir .....	24
2.3 Estructura lógica de la aplicación .....	25
2.3.2 Estilo de Arquitectura: Arquitectura en Capas.....	25

2.4 Framework: QT .....	27
2.4.1 Biblioteca Qwt .....	28
2.6 Lenguaje de Programación: C++ .....	29
2.8 Lenguaje de Modelado: UML .....	30
2.9 Metodología de Desarrollo: Proceso Unificado de Desarrollo (RUP).....	32
2.10 Herramienta Case: Visual Paradigm .....	33
2.11 Conclusiones Parciales.....	34
<b>Capítulo 3: Construcción y validación de la solución propuesta .....</b>	<b>35</b>
3.1 Introducción .....	35
3.2 Características del Sistema .....	35
3.2.1 Requisitos Funcionales .....	35
3.2.2 Requisitos No Funcionales.....	37
3.2.2.1 Usabilidad .....	37
3.2.2.2 Apariencia o Interfaz Externa .....	37
3.2.2.3 Rendimiento .....	38
3.2.2.4 Requerimiento de ayuda y documentación .....	38
3.2.2.5 Requisitos de Hardware .....	38
3.2.2.6 Portabilidad .....	38
3.2.3 Modelo de Casos de Uso del Sistema.....	38
3.2.3.1 Actores del Sistema .....	38
3.2.3.2 Casos de Uso del Sistema .....	39
3.3 Modelo de Diseño .....	41
3.3.1 Diagrama de Clases del Diseño.....	41
3.3.2 Patrones de diseño utilizados.....	44
3.3.2.1 Patrones GoF .....	44
3.3.2.2 Patrones GRASP .....	44
3.3.3 Descripciones de las clases.....	<b>¡Error! Marcador no definido.</b>

3.4 Estándar de Codificación .....	45
3.4.1 Declaración de variables.....	45
3.4.2 Espacios en blanco.....	46
3.4.3 Llaves .....	46
3.4.4 Paréntesis .....	48
3.4.5 Saltos de línea.....	48
3.4.6 Excepción General .....	48
3.5 Implementación.....	49
3.5.1 Modelo de Implementación .....	49
3.5.1.1 Vista Lógica .....	49
3.5.1.2 Vista de Implementación.....	50
3.5.1.3 Descripción de los componentes de la Vista de Implementación.....	50
3.5.1.4 Diagrama de componentes .....	51
3.6 Fase de Prueba .....	52
3.6.2 Pruebas de Caja Negra.....	53
3.6.2.1 Casos de Pruebas.....	53
3.7 Conclusiones Parciales.....	64
CONCLUSIONES GENERALES.....	65
RECOMENDACIONES .....	66
REFERENCIAS BIBLIOGRÁFICAS .....	67
GLOSARIO DE TÉRMINOS .....	70

**INDICE DE FIGURAS**

Figura 1 Gráfica de Barras ..... 8

Figura 2 Gráfica de Pastel ..... 9

Figura 3 Gráfico de área en 2D ..... 9

Figura 4 Gráfica de línea ..... 10

Figura 5 Gráfico de Superficie ..... 10

Figura 6. Distintas áreas del negocio ..... 13

Figura 7. Arquitectura en tres capas ..... 26

Figura 8 4+1 Vistas Arquitectónicas ..... 31

Figura 9. Ciclo de vida de RUP ..... 32

Figura 10 Diagrama de Casos de Uso del Sistema ..... 41

Figura 11 Diagrama de clases del diseño..... 44

Figura 12 Declaración de variables. Nombres cortos ..... 45

Figura 13 Declaración de variables. Abreviaturas ..... 46

Figura 14 Espacios en blanco. Palabras clave y llaves ..... 46

Figura 15 Espacios en blanco. Punteros o referencias ..... 46

Figura 16 Llaves. Clases y funciones ..... 47

Figura 17 Llaves. Condicionales ..... 47

Figura 18 Llaves. Instrucción de varias líneas..... 47

Figura 19 Llaves. Cuerpo vacío ..... 48

Figura 20 Paréntesis. Agrupar Expresiones ..... 48

Figura 21 Paréntesis. Operadores y comas ..... 48

Figura 22 Vista Lógica del Sistema..... 49

Figura 23 Vista General de Implementación ..... 50

Figura 24 Diagrama de Componentes ..... 52

Figura 25 Pruebas de Caja Negra..... 64

**INDICE DE TABLAS**

Tabla 1 Descripción de los actores del sistema ..... 38

Tabla 2 Clasificación de los Casos de Uso del Sistema ..... 39

Tabla 3 Caso de Prueba de la sección Nuevo Proyecto del caso de uso “Gestionar Proyecto” ..... 54

Tabla 4 Caso de Prueba de la sección Abrir Proyecto del caso de uso “Gestionar Proyecto” ..... 54

Tabla 5 Caso de Prueba de la sección Salvar Proyecto del caso de uso “Gestionar Proyecto” ..... 55

Tabla 6 Caso de Prueba de la sección Eliminar Proyecto del caso de uso “Gestionar Proyecto” ..... 56

Tabla 7 Caso de Prueba de la sección Exportar Ficheros con extensión .pdf del caso de uso “Exportar Ficheros” ..... 56

Tabla 8 Caso de Prueba de la sección Exportar Ficheros con extensión .jpg del caso de uso “Exportar Ficheros” ..... 57

Tabla 9 Caso de Prueba de la sección Importar Ficheros con extensión .las del caso de uso “Importar Ficheros” ..... 58

Tabla 10 Caso de Prueba de la sección Graficar Curvas del caso de uso “Construir Columna Curvas” ..... 59

Tabla 11 Caso de Prueba de la sección Insertar Litologías del caso de uso “Construir Columna Litológica” ..... 59

Tabla 12 Caso de Prueba de la sección Insertar Fósiles del caso de uso “Construir Columna Litológica” ..... 60

Tabla 13 Caso de Prueba de la sección Insertar Minerales del caso de uso “Construir Columna Litológica” ..... 61

Tabla 14 Caso de Prueba de la sección Insertar Texturas del caso de uso “Construir Columna Litológica” ..... 61

Tabla 15 Caso de Prueba de la sección Eliminar Símbolo del caso de uso “Construir Columna Litológica” ..... 62

Tabla 16 Caso de Prueba de la sección Nueva Descripción del caso de uso “Construir Columna Descripciones” ..... 62

Tabla 17 Caso de Prueba de la sección Modificar Descripción del caso de uso “Construir Columna Descripciones” ..... 62

Tabla 18 Caso de Prueba de la sección Eliminar Descripción del caso de uso “Construir Columna Descripciones” ..... 63

## INTRODUCCIÓN

El petróleo constituye hoy en día la fuente energética más usada (1), y por ende la más importante en el mundo. Se utiliza directa o indirectamente para casi todas las actividades que realizan los seres humanos. En la transportación se consume alguno de sus derivados como combustible, en el asfalto de las carreteras y en la fabricación de plásticos. En cada una de las pequeñas actividades diarias se está en contacto con alguna de sus aplicaciones. En cualquier momento si se analiza el entorno en el que se está, seguramente se podrán notar muchos elementos en las que directa o indirectamente está implicado este hidrocarburo (2).

Desde las primeras aplicaciones industriales del petróleo su importancia ha ido aumentando de manera gradual y ha alcanzado un ritmo tal que en los años venideros amenaza con quebrar el delicado equilibrio entre consumo y existencia. Los análisis de los expertos afirman que se agotarán las reservas existentes antes de lograr una solución alternativa al problema. Ante la inminencia que supone el lograr un balance o encontrar una fuente alternativa de energía que quizás no sea viable a corto y mediano plazo por costos, tecnología o cualesquiera otras razones, se hace necesaria una profundización radical y realista en las técnicas que hoy se aplican para obtener la fuente tradicional que es el petróleo. Por esta dependencia energética del hombre es precisamente que cada vez se dedican más esfuerzos, recursos en la investigación y el desarrollo de tecnologías para su extracción y tratamiento, así como para optimizar su utilización y aumentar su aprovechamiento. Actualmente miles de empresas, se dedican al enorme y creciente mercado del mundo del petróleo (2).

En la actualidad el uso de las Tecnologías de la Información y las Comunicaciones (TIC) en la Industria del Petróleo es de vital importancia para el desarrollo de nuestra sociedad. Con el desarrollo y el buen uso de las mismas se puede aumentar la producción de las empresas petroleras. La Industria del Petróleo en nuestro país, como en muchos otros, se divide en tres grandes procesos: Exploración-Producción, Refinación y Comercialización. Las entidades especializadas en Exploración-Producción, como su nombre lo indica, se encargan de la exploración y desarrollo de los campos, la perforación de nuevos pozos, así como la reparación de los que ya se encuentran en producción (3).

La Unión Cuba-Petróleo (CUPET) es la unión de empresas responsable del desarrollo de la industria del petróleo en Cuba, en conjunto con compañías extranjeras. A partir del incremento de la extracción de crudos nacionales, CUPET ha comenzado una fase de automatización de sus procesos en cada una de sus empresas y unidades, imprescindible para mejorar los resultados cualitativos y cuantitativos en cada uno de ellos. Como parte de CUPET existe el Centro de Investigaciones del Petróleo (CEINPET), en este se realizan actualmente la gran mayoría de las investigaciones relacionadas con el campo, único de su tipo en el país. Este centro está avalado por una alta calificación y experiencia de

trabajo, el cual se encarga de dar respuesta de forma integral a toda la actividad petrolera, desde la exploración hasta la refinación, en el proceso de investigación-desarrollo-producción.

Dentro del CEINPET radica el Departamento de Sedimentología, el cual se encarga de elaborar el Parte Diario de Geología; este incluye las columnas litológicas de todos los pozos en perforación, las cuales deben poder modificarse por el geólogo en cualquier momento que considere conveniente. La columna litológica no es más que una representación a escala de los componentes litológicos que están presentes en la columna de perforación que sigue el pozo en su camino hasta el hidrocarburo. Luego de ser confeccionado el Parte Diario de Geología, debe ser enviado a las altas esferas de la unión CUPET y del Ministerio de la Industria Básica (MINBAS).

En las plataformas de perforación el proceso para graficar las columnas litológicas está regido por rutinas que no permiten un rango muy amplio de modificaciones. Conectados a la plataforma de perforación hay una serie de sensores que obtienen una serie de datos muy valiosos, de esta forma se registra automáticamente, la velocidad de perforación, la rotación de la barrena y la profundidad tanto por el instrumento, como por la vertical al plano de la tierra. Las muestras de material geológico se extraen cada 5 metros y se trasladan a un laboratorio. Dentro del laboratorio se realiza un proceso de preparación de las mismas que consiste en lavarlas, tamizarlas, pesarlas y secarlas para poder ser analizadas con el calcímetro y el microscopio. Luego del análisis de estas muestras se obtienen datos que son representados mediante curvas y datos correspondientes a formaciones geológicas. En una misma muestra pueden darse más de un tipo de formación geológica, ante esta situación se define la que aparece en mayor abundancia y es ésta la que se grafica en la columna litológica.

En la Universidad de las Ciencias Informáticas (UCI), se desarrollan sistemas para la automatización de los procesos de la Industria del Petróleo. En la Facultad 5, se encuentra el Centro de Desarrollo de Informática Industrial, donde se desarrolla el Sistema de Manejo Integral de Perforación de Pozos (SIPP), el cual responde a las necesidades del CEINPET y de la Dirección de Intervención y Perforación de Pozos (DIPP), entidad responsable de todos los pozos en perforación en tierra del territorio cubano. En el proyecto SIPP se desarrollan sistemas (componentes o aplicaciones) que tienen como principal característica un gran procesamiento y rendimiento visual, debido a que se necesitan representar gran cantidad de datos en formatos gráficos.

Actualmente no existe en el CEINPET ningún sistema que se encargue de graficar las columnas litológicas de los pozos de petróleos. Este proceso se confecciona manualmente utilizando el Paint para ir graficando poco a poco cada tipo de litología presente en las distintas capas de material por las que atraviesa el pozo en perforación. En algunos casos utilizan software que proveen compañías extranjeras, como es el Strip Log del paquete de software Wellsight System. Estos software están

implementados en el sistema operativo Windows perteneciente a la empresa Microsoft. Además estos sistemas trabajan bajo licencias privativas, lo que unido al difícil acceso de las actualizaciones, por su costo, provoca que parte de los trabajadores que laboran con los mismos se vean excluidos de sus principales funcionalidades.

La utilización de software que trabajan bajo licencias privativas, en las entidades de la industria del petróleo de nuestro país, constituye una gran desventaja a la hora de procesar los datos obtenidos de los pozos de petróleo. Esto se fundamenta con el inconveniente que tienen estos software al no poderse realizar modificaciones sobre los mismos, debido a que no se puede tener acceso al código fuente, por lo cual no se pueden adaptar a la realidad de cada medio donde se use. Los especialistas del CEINPET requieren de un software que no esté atado a las restricciones de las aplicaciones privativas que se usan en las plataformas de perforación.

Teniendo en cuenta la situación problemática planteada se hace necesario implementar un sistema que permita graficar la columna litológica de los pozos en perforación, que disminuya el costo de tiempo y esfuerzo, y que simplifique el proceso de representación gráfica. Para ello se ha identificado como **problema a resolver** ¿Cómo mejorar la automatización del proceso de representación gráfica de columnas litológicas de los Pozos de Petróleo?

Para darle solución a este problema se realiza un análisis sobre el proceso de representación gráfica de columnas litológicas, lo que constituye el **objeto de estudio** de este trabajo, por lo que se define como **objetivo general** implementar el Sistema para Graficar Columnas Litológicas de Pozos de Petróleo, con el objetivo de mejorar la automatización del proceso de representación gráfica de columnas litológicas, enmarcado en el **campo de acción** Representación Gráfica de Columnas Litológicas del Sistema de Manejo Integral de Perforación de Pozos (SIPP).

Se defiende la idea de que implementando un Sistema para Graficar las Columnas Litológicas de Pozos de Petróleo se logrará mejorar la automatización del proceso de representación gráfica de columnas litológicas. Para lograr exitosamente la construcción de éste sistema se han planificado las tareas de la investigación que se relacionan a continuación:

1. Realización de un estudio del estado del arte para conocer el alcance de los sistemas de representación de columnas litológicas.
2. Descripción de las herramientas que se utilizarán para el desarrollo del Sistema.
3. Implementación del Sistema.
4. Validación de la Solución.

El **resultado que se espera** de este trabajo es un Sistema para Graficar Columnas Litológicas de Pozos de Petróleo, que permita a los trabajadores del CEINPET realizar el proceso de representación gráfica de una forma sencilla y eficiente, ahorrándole a la empresa tiempo y dinero de inversión en otros software propietarios e inapropiados. Para la realización de las tareas de investigación se han empleado los métodos de investigación que se describen a continuación.

**Métodos Teóricos:** Permiten conocer las relaciones que fluyen alrededor del objeto de estudio.

- ✓ **Inductivo-Deductivo** se utilizará este método para darle solución al problema planteado a través de conocimientos generales sobre el tema de investigación, llegando así a conclusiones particulares.
- ✓ **Histórico-lógico** es utilizado para analizar las soluciones existentes vinculadas al campo de acción. Se recopila información referente a las principales empresas del área y las herramientas que usan para lograr sus objetivos puesto que son muy similares a los que se quieren lograr al proveer al CEINPET de esta herramienta.

**Métodos Empíricos:** Permiten la observación y el análisis inicial de la información.

- ✓ **Observación**, con el objetivo de obtener un registro visual de toda la información referente a los procesos de perforación y el funcionamiento del centro en una situación real.

### **Estructura del contenido:**

Para un mejor entendimiento del trabajo se decide estructurar el mismo de la siguiente manera:

- ✓ **Capítulo 1.** “Fundamentación Teórica”: Se abordan los conceptos asociados al problema, al mismo tiempo se describe el objeto de estudio y la situación problemática. Se presenta la descripción del estado de arte con las soluciones que existen alrededor del objeto de estudio. Además se describen los distintos aspectos de la programación que son utilizados en la realización de este trabajo.
- ✓ **Capítulo 2.** “Herramientas y tecnologías actuales a utilizar”: Se abordan las herramientas y tecnologías que van a ser utilizadas en el desarrollo del sistema. Se describen los aspectos principales sobre el tipo de aplicación a construir y la estructura lógica que tendrá el sistema.
- ✓ **Capítulo 3.** “Construcción y validación de la solución propuesta”. En este capítulo se redefine el modelo de diseño del sistema, debido a los cambios sufridos en la etapa de implementación. Se especifica el estándar de codificación utilizado en el desarrollo del sistema y se describen los

atributos, métodos y fragmentos de códigos de las principales clases. Además se realizan las pruebas de caja negra, con el objetivo de probar el funcionamiento final de la aplicación.

### CAPÍTULO 1: Fundamentación Teórica

#### 1.1 Introducción

En este capítulo se abordan aspectos fundamentales sobre la representación gráfica, así como la importancia de la misma en el área de perforación de pozos de petróleo. Se definen los conceptos asociados al dominio del problema y se reseñan brevemente algunos software utilizados por los especialistas del CEINPET. Se describen los distintos tipos de gráficos que utilizan los software de representación gráfica y su correspondencia con las etapas del proceso de Perforación Petrolera. Se aborda de forma detallada el objeto de estudio y la situación problemática. También se analizan los principales software generadores de reportes gráficos que existen actualmente.

#### 1.2 Conceptos asociados al dominio del problema

Para lograr una mejor comprensión y debido a que el dominio del problema que se le da solución en este trabajo es complejo y requiere de mucha experiencia para poder conceptualizar los términos y conceptos que se manejan, es necesario definir algunos conceptos que no necesariamente deben formar parte del fondo de conocimiento de personas interesadas en este trabajo.

**Petróleo:** El petróleo es una mezcla heterogénea de compuestos orgánicos, principalmente hidrocarburos insolubles en agua, que se extrae de lechos geológicos continentales o marítimos. También es conocido como petróleo crudo o simplemente crudo. Es un líquido aceitoso, inflamable, cuyo color varía de incoloro a negro (4).

**Pozo de Petróleo y Gas:** Entidad donde se realiza la perforación, la cual agrupa un gran número de entidades y trabajadores, como son las compañías de servicio de lodo, direccionales, contratistas, logísticas, entre otras (4).

**Industria del Petróleo:** La industria petrolera incluye procesos globales de exploración, extracción, refino, transporte (frecuentemente a través de buques petroleros y oleoductos) y mercadotecnia de productos del petróleo (4).

**CUPET:** Unión de Empresas de Cuba encargadas de toda la actividad petrolera desde la exploración hasta la refinación, así como de satisfacer las necesidades del mercado nacional de hidrocarburos, a partir del incremento de la extracción de crudos nacionales (5).

**Procesos en la Industria del Petróleo:** La industria del petróleo se divide en 5 procesos fundamentales: Exploración, Perforación, Producción, Refinación y Comercialización. Estos cinco procesos sustentan todos los procesos que se realizan en la industria que tiene como resultado final

los derivados del hidrocarburo. Existen varios métodos para realizar la exploración, actualmente la más usada es la exploración magnética que se basa en el envío y recepción de ondas a través de la superficie terrestre. La perforación más usada en Cuba es la direccional que se trata de perforar el caño hasta aproximadamente 200 metros y darle un ángulo para buscar el reservorio horizontalmente, se usa de tierra al mar. La producción fundamental es la surgente o por bombas de agua, la surgente es en la que el petróleo sale sin ayuda, la otra se realiza inyectando agua en el reservorio y como el petróleo es menos denso sale a la superficie. La refinación se realiza en plantas como la Níco López, se trata de separar el petróleo de otros componentes como el azufre y ácidos y otros minerales. La comercialización la realiza CUPET en más de 200 puntos de venta de gasolina y diesel distribuidos en todo el país, así como en la generación de electricidad donde se usa la mayoría del petróleo extraído en Cuba (4).

El petróleo se almacena en los intersticios<sup>1</sup> que hay entre las partículas de los sólidos de estructura discontinua, lo que constituye una cualidad que poseen las rocas que permite el almacenamiento del petróleo, denominada **Porosidad**. Con el objetivo de determinar la existencia de esta cualidad en las rocas se lleva a cabo la **Perforación**, proceso de la Industria del Petróleo mediante el cual se logra acceder al mismo (2).

**Calcimetría:** Esta palabra no está registrada en el diccionario de la Real Academia Española (RAE<sup>2</sup> a partir de ahora). El término es usado por los especialistas del Departamento de Sedimentología para definir la cantidad de cal contenida en la muestra mediante un instrumento denominado **Calcímetro** (2).

**Geología:** Ciencia que trata de la forma exterior e interior del globo terrestre, de la naturaleza de las materias que lo componen y de su formación, de los cambios o alteraciones que estas han experimentado desde su origen, y de la colocación que tienen en su actual estado (6).

**Sedimentología:** Esta palabra no está registrada en el diccionario de la RAE. Sin embargo realizando un análisis se llega a la conclusión de que es una palabra compuesta por la palabra **sedimento** del latín *sedimentum*, definido como la materia que, habiendo estado suspensa en un líquido, se posa en el fondo por su mayor gravedad, y el elemento compositivo: **logía**, el cual significa 'tratado', 'estudio' o 'ciencia'. Si se realiza un análisis individual de las mismas se puede concluir como una definición aceptable de sedimentología el estudio y análisis de los sedimentos (2).

**Litología:** Del griego: λιθολογία. Parte de la geología que trata de las rocas (6), especialmente de su tamaño de grano, del tamaño de las partículas y de sus características físicas y químicas. Es parte de

---

<sup>1</sup> Espacios huecos.

<sup>2</sup> Siglas para Real Academia Española.

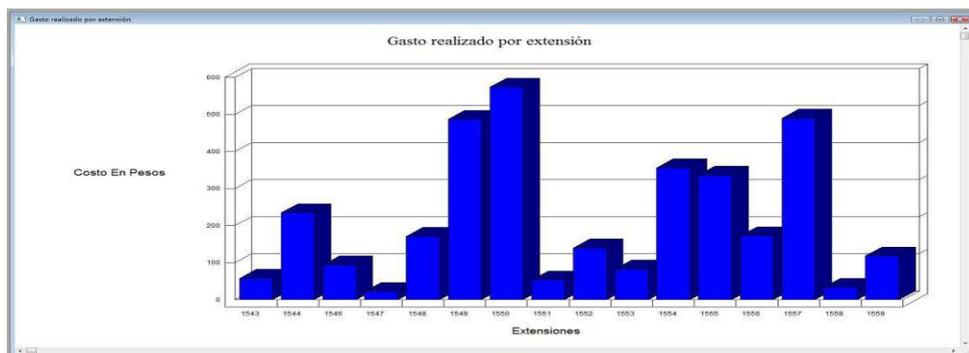
una terminología muy usada en el mundo del petróleo puesto que del estudio de las rocas depende en gran medida muchas de las acciones en la exploración, perforación y extracción del mismo (2).

**Gráfica:** Una gráfica es la representación de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que esos datos guardan entre sí. También puede ser un conjunto de puntos, que se plasman en coordenadas cartesianas, y sirven para analizar el comportamiento de un proceso, o un conjunto de elementos o signos que permitan la interpretación de un fenómeno (7).

### 1.3 Tipos de gráficos en perforación

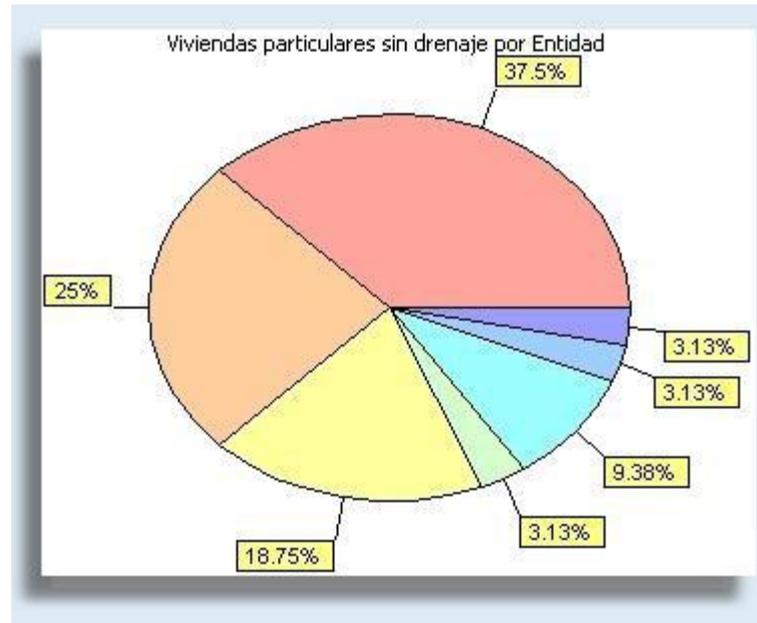
Existen empresas que realizan software para la Industria del Petróleo con representaciones gráficas de gran calidad y precisión, como la compañía Schlumberger que emplea el paquete de Software WellWizard utilizado en los centros petroleros cubanos. En estos software abunda mucho el uso de visualizaciones utilizando gráficos. Es muy amplia la diversidad de gráficos existentes. La utilización de cada uno de ellos depende de los diferentes conjuntos de datos y del uso para el que estén confinados. A continuación hay una vista general de los tipos de gráficos principales y sus usos más comunes.

- ✓ Gráficos de Barras: También se le llama gráfico de columnas, es útil para presentar o comparar varios conjuntos de datos, en el área de perforación se utilizan para mostrar cambios de datos en un período de tiempo o para ilustrar comparaciones entre elementos.



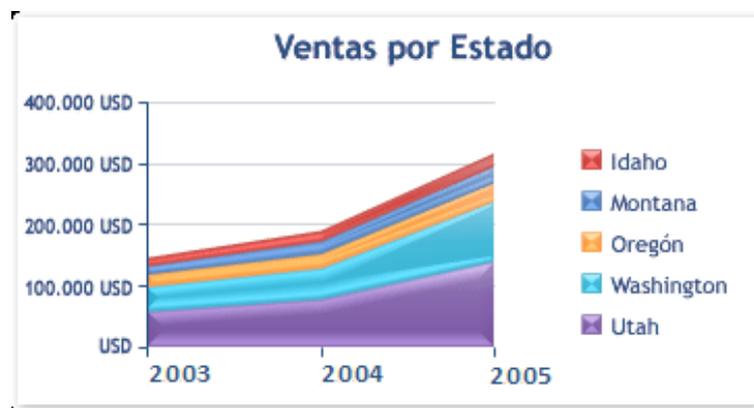
**Figura 1** Gráfica de Barras

- ✓ Gráficos de Pastel: Un gráfico de pastel muestra los datos como un círculo dividido en secciones de colores o diseños, en la industria petrolera se utiliza mucho para representar un grupo de datos (por ejemplo el porcentaje de la distribución del tiempo de perforación de un pozo).



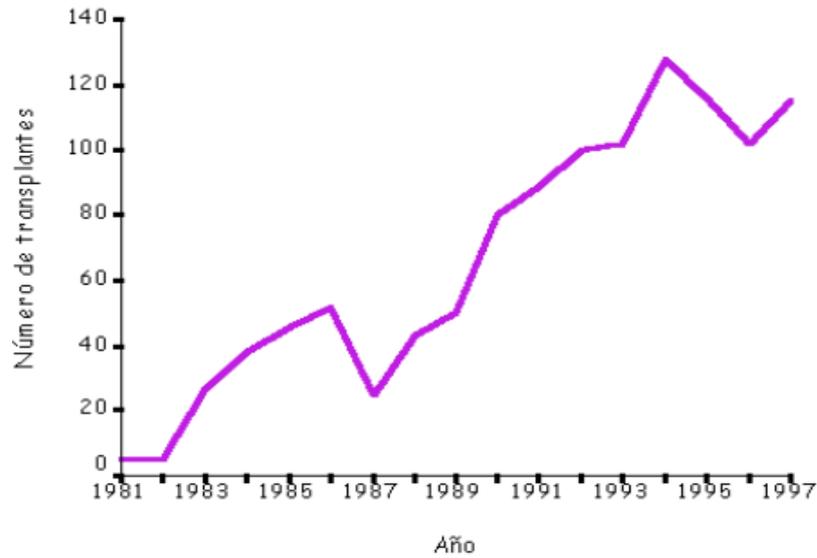
**Figura 2** Gráfica de Pastel

- ✓ Gráficos de Área: Cuando se hace necesario mostrar la variación de una o varias variables en correspondencia con el tiempo, se acude en los software de perforación, a dichos gráficos. Es común ver su uso con abundantes colores, donde a cada uno de ellos le corresponde una variable.



**Figura 3** Gráfico de área en 2D

- ✓ Gráficos de Líneas: Este tipo de gráfico es una serie de puntos conectados entre sí mediante rectas, donde cada punto puede representar distintos valores, los cuales son representados en dos ejes cartesianos ortogonales entre sí. Son usados cuando se está en presencia de una variable en función del tiempo, siendo más comprensible la variación que un gráfico de área.

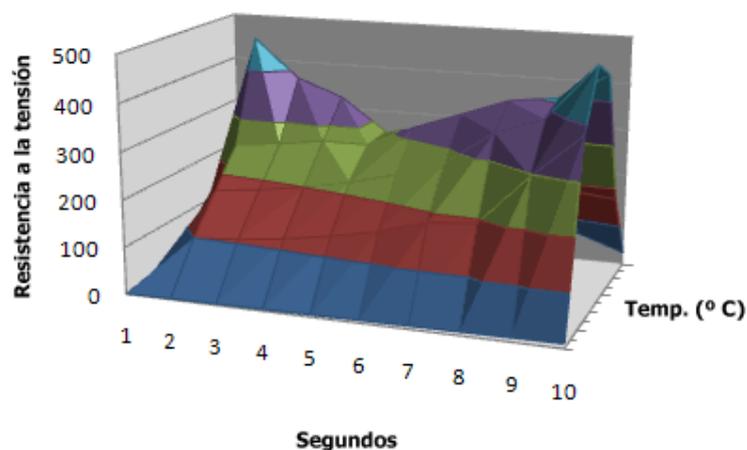


**Figura 4** Gráfica de línea

Luego de analizadas las características de este tipo de gráfico, se selecciona como el adecuado para representar en el sistema los datos que necesiten ser graficados mediante curvas, por ejemplo, los gases y la porosidad.

- ✓ Gráficos de Superficie y Gráficos de Áreas en 3D: Estos gráficos son muy utilizados en la presente área ya que son capaces de representar determinadas características geológicas que favorecen cuando se va a perforar. Un ejemplo lo constituyen las características de las capas del suelo donde se realizará la perforación.

**Mediciones de resistencia a la tensión**



**Figura 5** Gráfico de Superficie

### 1.4 Representación gráfica de Columnas Litológicas

#### 1.4.1 Descripción general

El proceso de construcción de columnas litológicas en el departamento de Sedimentología del CEINPET pudiera definirse como el proceso que se lleva a cabo para analizar la trayectoria de los datos a graficar de los pozos de petróleo en perforación. A la representación gráfica de estos resultados se le llama Columna Litológica. Para poder realizar los análisis necesarios, comparaciones y una buena interpretación, es necesario poner estos datos en un formato gráfico que constituya una aproximación a escala de la realidad, donde se pueda apreciar los diferentes tipos de litologías que atraviesa el pozo en su camino hacia el hidrocarburo. La elaboración de estas gráficas es una de las tareas que realizan los trabajadores del Departamento de Sedimentología del CEINPET.

A partir del análisis de la información contenida en estas gráficas se podrá arribar a conclusiones que favorezcan las decisiones que se tomarán posteriormente en la inversión de un pozo petrolero; si es factible perforar un nuevo pozo, hasta qué profundidad se ha perforado y cuántos días han demorado en alcanzar esa profundidad. Para lograr una representación lo más cercana a la realidad y un buen entendimiento de la misma se establecen una serie de pautas a seguir:

- ✓ Debe estar precedida por el encabezamiento, una serie de datos específicos de cada pozo: localización, estado, fecha de inicio de perforación. Estos datos no son presentados como contenido de este trabajo, debido a su carácter sensible, lo cual viola las disposiciones legales autorizadas para el uso de datos en trabajos de este tipo. Se incorporarán luego de almacenada la información no sensible dependiendo del uso que se le vaya a dar a la misma y siempre por personal competente autorizado.
- ✓ En la parte superior debe poseer un menú y una barra de herramientas, donde se podrá acceder a las principales funcionalidades del software, además de los eventos del mouse en el área de trabajo.
- ✓ El área de trabajo debe estar dividida en columnas donde cada una representará un campo, quedando como sigue: Columna Profundidad, Columna Curvas, Columna Litológica y Columna Descripción.
  - La **Columna Profundidad** es la columna donde se ubican los datos de profundidad del pozo. Existen dos tipos de profundidad, Measured Depth (MD) o profundidad de la medición y la True Vertical Depth (TVD) o profundidad vertical real. Estas especificaciones se deben a que los pozos petroleros en perforación rara vez son perforados en línea recta por la vertical al plano de la tierra sino que presentan desviaciones, incluso curvas pronunciadas dependiendo del

estudio preliminar del terreno, la estrategia a seguir, las rupturas o interrupciones que puedan surgir en la perforación y otros factores de consideración. Para la realización de esta herramienta se utilizará la MD, puesto que es la profundidad que sigue el pozo sea cual fuere la dirección de este.

- En la **Columna Curvas** estarán representadas una serie de gráficas que se muestran con curvas como son: la Velocidad de Perforación, Impregnación, Porosidad, Calcimetría y los Gases.
  - ✓ La **Velocidad de Perforación** se mide en minutos por metro o en metros por hora dependiendo de si es una baja, media o alta velocidad. Debe acordarse previamente la unidad de medida con el geólogo de operaciones. La velocidad de perforación es la velocidad a la que avanza la perforación del pozo en cada uno de los tramos que lo conforman. Como norma se colocan los valores máximos a la izquierda, de forma tal que la curva se desplace hacia la izquierda cuando los valores de la velocidad aumenten y viceversa.
  - ✓ La **Impregnación** es un parámetro para medir el grado de impregnación de hidrocarburo en la roca. Se evalúa cualitativamente en 4 indicadores de calidad: Pobre, Buena, Excelente o No Existe.
  - ✓ La **Porosidad** representa la distribución de los poros en la roca. Está directamente relacionado con la capacidad de impregnación de la misma. Se evalúa cualitativamente atendiendo al análisis microscópico de la muestra en 5 indicadores de calidad: Muy bien seleccionada, Bien seleccionada, Moderadamente seleccionada, Pobremente seleccionada o Muy Pobremente seleccionada.
  - ✓ La **Calcimetría** responde al análisis del calcímetro, es la cantidad de cal contenida en la muestra. Estos datos se recogen automáticamente en la plataforma de perforación.
  - ✓ Los **Gases** se miden en por ciento. Para representarlos gráficamente en una sola curva se promedian los por cientos de los 8 tipos de gases comunes que se obtienen en las trampas de gas.
- La **Columna Litológica** es donde se representa gráficamente los diferentes componentes litológicos presentes en las capas de material por las que atraviesa el pozo.
- La **Columna Descripción** se refiere a la descripción de los tipos de litología. Se hace necesaria dado que existen especificidades o singularidades que deben ser señaladas

mediante anotaciones puesto que la biblioteca de símbolos no las recoge. Es la forma en que los especialistas pueden hacer de manera rápida anotaciones al margen de cada una de las secciones que están analizando (2).

## 1.4.2 Descripción actual del dominio del problema

Actualmente en el CEINPET, específicamente en el Departamento de Sedimentología, para graficar la columna litológica de los pozos en perforación no se usa un software único que proporcione estabilidad y estandarización en las prácticas laborales. Esta situación está dada por múltiples factores, por ejemplo, no existe una única compañía trabajando sobre el mismo terreno. Las diferentes compañías extranjeras que trabajan en conjunto con los especialistas cubanos en el campo usan cada una el software con el cual trabajaban anteriormente, al cual deben forzosamente adaptarse los técnicos y trabajadores del CEINPET.

El entorno del negocio de las entidades responsables de los pozos en perforación se encuentra dividido en tres áreas fundamentales (8):

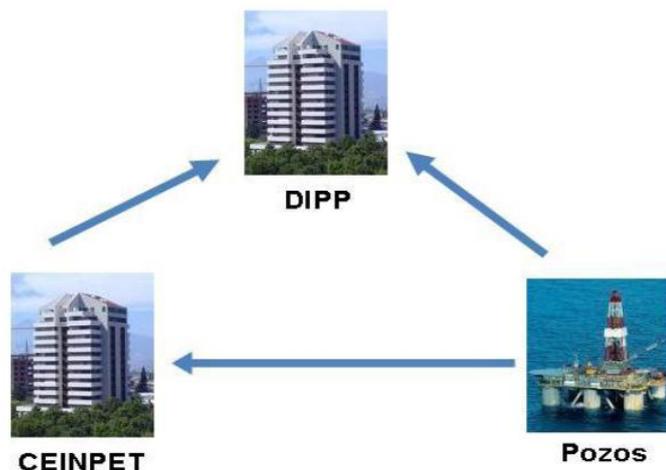


Figura 6. Distintas áreas del negocio

**DIPP:** Se encuentra ubicada en el municipio Varadero perteneciente a la provincia de Matanzas, aquí se confeccionan una serie de informes, partes y reportes a mano con el software Microsoft Office Excel (Programa perteneciente al paquete de Office creado por la Microsoft) con las informaciones que se reciben diariamente a las 8:00 de la mañana por correo electrónico de los pozos petroleros en perforación y del CEINPET, en caso de no existir la conexión la misma es gestionada por teléfono. Estos documentos generados contienen información de la perforación diaria, donde se incluyen informaciones de la cantidad de combustible, barrenas, herramientas, camisas y productos químicos utilizados en los pozos, así como las operaciones y actividades que se realizarán en el día, además de la cantidad de dinero invertido. Los mismos son manipulados por las secretarías de la DIPP, aprobada

por el Jefe de Despacho, y guardada en formato duro y digital para tomar decisiones en momentos posteriores (8).

**CEINPET:** Pertenece a la Unión CUPET, la misma está localizada en Boyeros, provincia Ciudad Habana. Aquí diariamente la información es recibida en forma de reporte a las 7:00 de la mañana por correo electrónico al igual que en la DIPP. En caso de no existir conexión la información es gestionada por teléfono. La elaboración de los reportes se realiza de forma manual, utilizando el software Microsoft Excel para realizar el parte diario de geología y el Paint para dibujar poco a poco la columna litológica, la cual contiene todos los datos de las litologías de los pozos en perforación. Luego de recibir los reportes, son guardados en formato duro y digital, llevando de esta forma un historial con todos los documentos realizados (8).

**Pozos Petroleros en Perforación:** Están ubicados por toda la parte norte de nuestro país principalmente en las provincias de La Habana y Matanzas. En los mismos laboran una serie de personas que contribuyen a la gestión de la información generada en cada uno de los pozos, las cuales se describen a continuación (8):

- ✓ **Geólogo:** Realiza diariamente un estudio consecuente de las características de las rocas, del terreno, analizando todo tipo de formación, obtiene información del WellWizard, elabora el Reporte Diario de Geología, lo guarda en formato duro y digital y se lo entrega al Supervisor de Pozo, además de enviárselo al geólogo que trabaja en la oficina del CEINPET (8).
- ✓ **Supervisor de Pozo:** Recibe diariamente información en formato duro y digital del químico, geólogo, direccional y analiza minuciosamente el software WellWizard. Examina los parámetros recibidos, confeccionando con estos datos una serie de reportes. Luego de confeccionados, estos reportes se guardan en formato duro y digital y se envían diariamente a las 12:00 de la noche a la DIPP y al CEINPET (8).

### 1.5 Análisis de las soluciones existentes vinculadas al campo de acción

En la actualidad no existe una solución que resuelva totalmente el problema planteado en este trabajo. Las soluciones existentes lo resuelven parcialmente. Se tomarán algunas de ellas para realizar un análisis de sus principales características identificando debilidades y fortalezas en las mismas que servirán de base a la propuesta para desarrollar.

### **1.5.1 Descripción General**

Las gráficas que se generan en el proceso de construcción de las columnas litológicas son utilizadas en muchas operaciones como son: análisis de datos, estudios comparativos, exploración, análisis de riesgos y otras. Poseer un software que no cubra las necesidades para llevar a cabo esta tarea, al cual no se le puedan hacer modificaciones para mejorar sus funcionalidades y adaptarlo a la realidad de cada medio donde se use, es una desventaja seria para cualquier empresa del campo. Los software más usados en la actualidad son los que proveen las compañías rectoras a nivel internacional como la Schlumberger, la M-Tech Software, y muchas otras nuevas empresas como la Petrokem Loggin Services. Todos poseen como denominador común su implementación en Sistemas Operativos privativos y que pertenecen a empresas propietarias.

El Departamento de Sedimentología del CEINPET no cuenta con una aplicación que responda satisfactoriamente a las necesidades de los trabajadores del mismo y esto incide negativamente en el alcance de su trabajo. Los software usados en las diferentes plataformas de perforación cumplen con sus objetivos para el trabajo en las mismas, pero las necesidades varían desde el momento en que los trabajadores del Departamento de Sedimentología requieren modificar el contenido de los reportes. En las entidades de la Industria Petrolera de nuestro país en ocasiones se trabaja con un conjunto de software, entre ellos WellWizard y el Strip Log 6, que se encargan de interpretar y graficar los registros de pozos. Estos software trabajan bajo licencias privativas, lo que unido al difícil acceso de las actualizaciones, por su costo, provoca que parte de los trabajadores que laboran con el mismo se vean excluidos de las principales funcionalidades que brindan.

Entre los directivos de la DIPP la información se maneja vía correo electrónico y sin un diseño estipulado, por lo que al recibirla necesita ser reescrita varias veces. Esto provoca que se duplique el esfuerzo y es propicio para que se introduzcan errores humanos en los datos. Los supervisores reciben la información de los reportes por vía telefónica y se almacenan en hojas de papel, además, estos no siempre se almacenan cuando un pozo termina su estado de perforación.

Los especialistas para realizar su trabajo en el Departamento de Sedimentología del CEINPET requieren de un software que no esté atado a las restricciones de las aplicaciones privativas que se usan en las plataformas de perforación. Por tal razón se plantea la necesidad de desarrollar una aplicación que permita mejorar la automatización del proceso de representación gráfica de las columnas litológicas de los pozos de petróleo.

### 1.5.2 WellWizard

Sistema Integrado de Datos (WellWizard), es un sistema versátil que utiliza cualquier servicio de flujo de datos de pozo. Todos los datos geológicos y de perforación están integrados, en tiempo real, a través de la interfaz gráfica WellWizard.

WellWizard IDS es configurable para integrar datos de múltiples fuentes, incluyendo la perforación direccional, de medición durante la perforación, el registro durante la perforación y de ensayo y de producción. Posee un Sistema de Archivo que puede crear reportes de barras, mixtas y registros geológicos que se imprimen en papel continuo o guardan en archivos PDF. Es un sistema de muestreo de todos los factores que son fundamentales en la perforación de pozos como aire comprimido, temperatura de lodo, los cuales son monitoreados por un software que está actualmente instalado en todos los pozos en perforación de Cuba (9).

Ventajas (9):

- ✓ Más de 75 000 bombas de Well Wizard asistentes de la cámara de aire están en uso más que todas las otras marcas y tipos de muestras de agua y tierra combinados.
- ✓ Modelos para cada pozo, bajo rendimiento, corta columna de agua, a profundidades de 1.000 pies (304.8 m).
- ✓ Confiabilidad probada desde 1982, con el primer estándar de la industria de 10 años de garantía.
- ✓ Exclusiva formulación de la Cámara de aire de PTFE clasificado para muchos años de vida, mayor flexibilidad de los materiales de la de la misma.

### 1.6.3 Wellsight System

En el CEINPET, para el análisis de los datos y la construcción de las columnas litológicas, se usa actualmente un software propietario que provee la compañía Wellsight Systems, el Strip Log 6 (10). El trabajo con dicho software es eficiente porque permite graficar una serie de elementos necesarios para la visualización de los datos generados en la perforación, pero cuenta con una serie de desventajas. Para operar el Strip Log 6 es necesario poseer una licencia electrónica sin la cual no se puede acceder al software. Esto trae como consecuencia que exista un plan de contingencia única y exclusivamente para el caso de deterioro o extravío de dicha licencia.

Los datos que genera el software, si se quieren recibir en otro formato fuera del que se establece para un estudio más pormenorizado, tienen que modificarlos los trabajadores utilizando otros software. Al no tener privilegios no se puede acceder al código fuente y adaptarlo a las necesidades específicas de los geólogos cubanos. Esta aplicación provee una serie de utilidades que para el trabajo que se realiza en el CEINPET, son redundantes o superfluas, ya que no tienen uso definido dentro de las investigaciones que se llevan a cabo en el CEINPET.

### 1.5.4 WellFlo 3

WellFlo es una Suite de Programas que Modelan y Optimizan Pozos de Crudo, Gas y Redes de Producción para ingenieros petroleros que pronostiquen desempeños, diagnostiquen problemas de pozo u optimicen producciones desde instalaciones existentes. La versión 3 de WellFlo brinda un nuevo nivel de sofisticación, velocidad y precisión. Al mismo tiempo, el programa permanece suficientemente simple de usar, asegurando un aprendizaje rápido y un alto nivel de productividad, factor importante en la altamente exigente industria del petróleo y gas de hoy (11).

A continuación un resumen de las principales características de modelación de pozos que posee el software:

#### **Modelado de Pozos y realización de Pozos Horizontales (11).**

- ✓ Modelado preciso de pozos horizontales.
- ✓ Opciones de influjo de estado semi-permanente o permanente.
- ✓ Modelado de pérdida de presión a lo largo de la sección horizontal.
- ✓ Múltiples puntos de influjo.
- ✓ Modelado de múltiples huecos de drenaje.

#### **Salida Gráfica Disponible para Cálculos de Caída de Presión (contra profundidad vertical medida o real) (11).**

- ✓ Perfil a lo largo del pozo para Presión, Temperatura; tasas de flujo de Fase; gradientes de presión total, gravitacional, de fricción y aceleración; retenciones de fase “No slip”; velocidad de descarga de Turner; velocidades de fase n-situ y superficiales.
- ✓ La data medida de presión y temperatura puede ser graficada.

### 1.5.5 Strater

Strater es una potente e innovadora herramienta para el registro de pozos y el trazado de perforaciones. Con esta herramienta los geólogos ya no tendrán que invertir más tiempo y dinero para crear registros profesionales de pozos. Strater es una herramienta potente y sencilla de utilizar (12).

[\(Anexo 1\)](#)

Con Strater se puede visualizar gráficamente (12):

- ✓ Profundidad o elevación.
- ✓ Notas, comentarios y otros datos de texto.
- ✓ Petrología, por ciento de petrología y descripciones litológicas.
- ✓ Potencial espontáneo, rayos gamma, calibrador, neutrónica, densidad aparente, resistividad, ritmo de perforación, gas total, calidad de los gases y datos sísmicos.

- ✓ Contabilidad de impactos, número y tipo de muestras, permeabilidad, RQD<sup>3</sup>, por ciento de recuperación.
- ✓ Concentración de contaminantes, contenido de humedad y detalles de construcción de pozos.
- ✓ Datos de ensayos, petrología de mineralización o alteración, valores BTU<sup>4</sup> y datos de contenido de cenizas.
- ✓ Virtualmente cualquier tipo de dato de profundidad o intervalo.

Strater ofrece una flexibilidad insuperable para el diseño y formateo de registros. Su avanzada interfaz de usuario permite que el diseño y la visualización de sus datos se realicen de forma sencilla. Strater incluye 13 tipos de registros muy usuales para visualizar sus datos gráficamente: profundidad, línea/símbolo, gráfico cruzado, petrología, barras de zona, barras, porcentajes, postes, postes por clase, gráficos, textos complejos, y registros de construcción de pozos. Cada uno de los registros puede ser modificado para ajustarse a las necesidades del usuario. Entre algunas de las personalizaciones posibles se pueden citar (12):

- ✓ Visualizar registros basados en profundidad o elevación.
- ✓ Visualizar profundidad y/o líneas de rejilla variable.
- ✓ Añadir barras de escala y títulos.
- ✓ Configurar diferentes estilos de líneas de contacto entre unidades litológicas.
- ✓ Utilizar curvas para crear perfiles litológicos como perfiles de desgaste.
- ✓ Crear registros continuos sencillos o de múltiples páginas.
- ✓ Formatear diferentes tipos de registro para obtener presentaciones lo más informativas posible.

Strater proporciona varias funcionalidades para simplificar la tarea de importar datos, crear el diseño de la perforación exacta que el usuario requiere y obtener la salida en el formato necesario, ya sea impreso o exportado a formato electrónico para incluir en un informe o presentación.

Hasta aquí se han analizado los software de más alto prestigio en cuanto a la informática vinculada a la Industria Petrolera que resuelven parcialmente el problema que origina la realización de este trabajo. Todos tienen inconvenientes en común que conllevan a que estos software no sean los más idóneos a la hora de trabajar en las esferas relacionadas con un recurso tan importante como lo es el petróleo. El principal inconveniente es que todos trabajan bajo licencias privativas, por las cuales hay que pagar un costo muy elevado, además de que algunos serían imposibles de comprar debido a la política restrictiva de EE.UU implantada a nuestro país a través del Bloqueo Económico y Comercial.

---

<sup>3</sup> índice de calidad de las rocas.

<sup>4</sup> Unidades Térmicas Británicas.

En cuanto al producto que se desarrolla se pueden destacar unos cuantos aspectos donde se reflejan soluciones a los problemas presentados en los software analizados anteriormente. Este producto será completamente gratuito para CUPET, así como todas sus actualizaciones, además de que podrá adaptarlo a sus necesidades en caso de que se necesite realizar algún cambio en el software. Debido a que todos los procesos de negocio fueron levantados en un pozo de petróleo localizado en Matanzas, es el software que más se adapta a la industria petrolera nacional, el cual cumple con todas sus especificaciones.

## **1.6 Estándares de codificación**

Un estándar de codificación es una forma de "normalizar" la programación de forma tal que al trabajar en un proyecto, las personas involucradas en el mismo tengan acceso y comprendan el código. En otras palabras define la escritura y organización del código fuente de un programa. Por lo general los estándares de programación definen la forma en que deben ser declaradas las variables, las clases y los comentarios. Algunos estándares especifican qué datos del programador y de los cambios realizados al código fuente deben incluirse (4).

El código de un programa lo leen muchas personas, para repararlo, para ampliarlo o, simplemente, para evaluarlo. Para estos lectores es fundamental que el programa esté bien redactado, con estilo, para que su significado sea claro. Existen varias reglas básicas que ayudan a conseguir un texto satisfactorio (13).

## **1.7 Patrones**

Los Patrones son soluciones comunes a problemas de diseño de software orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables ya que pueden ser aplicados en otros diseños o problemas. Constituyen una disciplina problema-solución que está en constante evolución entre los diseñadores y desarrolladores que trabajamos con lenguaje orientado a objetos.

Un patrón es:

- ✓ La abstracción de una forma concreta que puede repetirse en contextos específicos.
- ✓ Una información que captura la estructura esencial y la perspicacia de una familia de soluciones probadas con éxito para un problema repetitivo que surge en un cierto contexto y sistema.
- ✓ Una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de un cierto contexto (14).

Los Patrones de diseño se clasifican en:

- ✓ Patrones de creación: Inicialización y configuración de objetos.
- ✓ Patrones estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases u objetos se agrupan, para formar estructuras más grandes.
- ✓ Patrones de comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos (15).

Aunque cuando se habla sobre los patrones salen a relucir los patrones de diseño como los más importantes en el mundo de la programación orientada a objetos, estos no son los únicos que existen. Actualmente hay una gran cantidad de patrones que sería difícil para un programador el tratar de conocerlos todos. A continuación se muestran algunos de ellos:

- ✓ **Patrones de Arquitectura:** Soluciones probadas para estructurar los componentes, como son Modelo-Vista-Controlador, arquitectura en tres capas, peer to peer, arquitectura orientada a servicios, por citar ejemplos.
- ✓ **Patrones Web:** Soluciones probadas para la creación de sitios web, como son la maquetación en tres columnas, efectos de rollover y estructuras de blog.
- ✓ **Patrones de Diseño:** Las soluciones probadas para el diseño de software.
- ✓ **Patrones de Programación:** Soluciones específicas para algoritmos y estructuras de control, como son algoritmos de ordenación, procedimientos de recursión e iteración, etc.
- ✓ **Patrones de Refactorización:** Soluciones para simplificar el código, como la variable explicativa, extracción de métodos, sobrecarga de constructores, etc. (3).

### 1.8 Estilos Arquitectónicos

Un estilo arquitectónico es una familia de sistemas de software en términos de su organización estructural. Expresa componentes y las relaciones entre estos, con las restricciones de su aplicación y la composición asociada, así como también las reglas para su construcción (16).

Los estilos son útiles para sintetizar estructuras de soluciones, también definen los patrones posibles de las aplicaciones y permiten evaluar arquitecturas alternativas con ventajas y desventajas conocidas ante diferentes conjuntos de requisitos no funcionales. Algunos de los estilos arquitectónicos más conocidos son:

- ✓ Estilos de Flujo de Datos
- ✓ Estilos Centrados en Datos
- ✓ Estilos de Llamada y Retorno
- ✓ Estilos de Código Móvil

- ✓ Arquitectura basada en objetos
- ✓ Estilos heterogéneos
- ✓ Estilos Peer-To-Peer
- ✓ Estilos derivados

### 1.9 Programación Orientada a Objetos (POO)

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

La programación orientada a objetos trata de amoldarse al modo de pensar del hombre y no al de la máquina. Esto es posible gracias a la forma racional con la que se manejan las abstracciones que representan las entidades del dominio del problema, y a propiedades como la jerarquía o el encapsulamiento (4).

La POO es un paradigma que utiliza objetos como elementos fundamentales en la construcción de la solución. Un objeto es una abstracción de algún hecho o ente del mundo real que tiene atributos que representan sus características o propiedades y métodos que representan su comportamiento o acciones que realizan. Todas las propiedades y métodos comunes a los objetos se encapsulan o se agrupan en clases. Una clase es una plantilla o un prototipo para crear objetos, por eso se dice que los objetos son instancias de clases.

#### 1.9.1 Características de la POO

Existe un acuerdo acerca de qué características contempla la "orientación a objetos", las características siguientes son las más importantes (4):

- ✓ **Abstracción:** Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar *cómo* se implementan estas características.
- ✓ **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

- ✓ **Modularidad:** Se denomina Modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Al igual que la encapsulación, los lenguajes soportan la Modularidad de diversas formas.
- ✓ **Principio de ocultación:** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos, que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado.
- ✓ **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama *asignación tardía* o *asignación dinámica*.
- ✓ **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo.
- ✓ **Recolección de basura:** la recolección de basura o *garbage collector* es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando.

### 1.9.2 Ventajas e inconvenientes de la P00

Entre las ventajas más importantes podemos destacar:

- ✓ Favorece la comunicación entre analistas, diseñadores, programadores y usuarios finales al utilizar todos los mismos modelos conceptuales.
- ✓ Esto se traduce en un aumento de la productividad, ya que la comunicación es uno de los puntos críticos en las primeras fases del proyecto.

- ✓ Se facilita la representación de estructuras complejas sin necesidad de adaptarnos a normas y modelos, ya que lo que manejamos son objetos del mundo real, lo que facilita la tarea del analista.
- ✓ La semántica de estas técnicas es más rica (al ser más natural); al usuario final le es más fácil comprender lo que el analista representa en sus modelos (ya que representa los objetos que lo rodean habitualmente).
- ✓ Favorece la modularidad, la reusabilidad y el mantenimiento del software.
- ✓ Estas técnicas son más resistentes al cambio que las tradicionales técnicas de análisis orientadas a flujos de datos.

Algunas de sus desventajas:

- ✓ Hay que ser muy cuidadosos en la creación de los objetos, ya que de ello dependerá el éxito de nuestro proyecto. Un error en estas primeras definiciones podría resultar catastrófico. Precisamente el secreto de esta técnica está en la correcta definición inicial de los objetos.
- ✓ Los estándares en este tipo de técnicas están en continua evolución, lo que exige una actualización permanente.
- ✓ Los analistas, diseñadores y desarrolladores del proyecto deben conocer las reglas del juego y poseer suficiente experiencia en programación.

### 1.10 Conclusiones Parciales

El estudio realizado sobre el objeto de estudio y la situación problemática proporcionaron un gran conocimiento y mejor entendimiento del problema a resolver con este trabajo.

Luego del estudio de las soluciones existentes se pudo afirmar que ninguna de ellas satisface en su totalidad las necesidades del CEINPET y permitió detallar las características esenciales que debe tener el sistema.

El análisis de los diferentes tipos de gráficos que son utilizados en el proceso de visualización gráfica de distintas empresas que realizan software para la Industria del Petróleo, permitió la selección de los gráficos más idóneos a utilizar en el sistema que se desarrolla.

### Capítulo 2: Herramientas y tecnologías actuales a utilizar

#### 2.1 Introducción

En el presente capítulo se abordan los aspectos fundamentales relacionados con la arquitectura de software propuesta para la elaboración del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo. Además se describen las descripciones, ventajas y desventajas de cada una de las herramientas y tecnologías a emplear para una buena elaboración del software, que garantice como resultado final un producto con la calidad requerida.

#### 2.2 Tipo de Aplicación a construir

Para elegir un tipo de aplicación se debe tener presente las restricciones de despliegue, de conectividad, de lo compleja que sea la interfaz de usuario y de las restricciones de interoperabilidad, flexibilidad y tecnologías que imponga el cliente (17).

En la captura de requisitos realizada en conjunto con el cliente, teniendo en cuenta las características del proceso de Graficar Columnas Litológicas y el ambiente de uso de la aplicación, se identificaron una serie de características que deben estar presentes en el sistema a desarrollar. La aplicación debe tener una gran capacidad de procesamiento y rendimiento visual. Las aplicaciones de escritorio brindan facilidades específicas y prestaciones para la representación de gráficos, importar y exportar documentos, así como para gestionar todo el procesamiento de ficheros, tareas necesarias para el desarrollo del sistema requerido. Por estas razones se ha escogido una aplicación de escritorio, ya que la misma cuenta con las características antes mencionadas. Esta selección se fundamenta también en el hecho de que el ambiente de despliegue de la aplicación no está definido; esto significa que los geólogos puedan graficar las columnas en cualquier lugar, CEINPET, pozos o su propia casa, por tanto una arquitectura web donde se depende de la conexión y de un servidor no cumple estas especificaciones. Teniendo en cuenta el análisis realizado se pueden definir las siguientes ventajas de las aplicaciones de escritorio: (17)

- ✓ Aprovechan mejor los recursos de los clientes.
- ✓ Ofrecen la mejor respuesta a la interacción, una interfaz más potente y mejor experiencia de usuario.
- ✓ Proporcionan una interacción muy dinámica.
- ✓ Soportan escenarios desconectados o con conexión limitada.

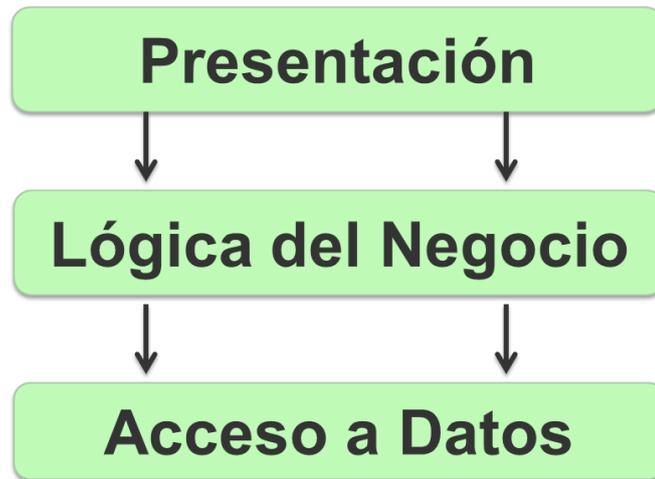
### 2.3 Estructura lógica de la aplicación

“Para lograr un buen entendimiento de la estructura lógica de la aplicación se realiza una descripción sobre las capas (layers), así como los beneficios que trae el uso de las mismas. Las capas son agrupaciones horizontales lógicas de componentes de software que forman la aplicación o el servicio. Nos ayudan a diferenciar entre los diferentes tipos de tareas a ser realizadas por los componentes, ofreciendo un diseño que maximiza la reutilización y especialmente el mantenimiento. El dividir una aplicación en capas separadas que desempeñan diferentes roles y funcionalidades, nos ayuda a mejorar el mantenimiento del código; nos permite también diferentes tipos de despliegue, y sobre todo, nos proporciona una clara delimitación y situación de dónde debe estar cada tipo de componente funcional e incluso cada tipo de tecnología” (17).

#### 2.3.2 Estilo de Arquitectura: Arquitectura en Capas

Con el estudio realizado se llega a la conclusión de que el estilo arquitectónico que más se adapta a la solución es la Arquitectura en Capas. La utilización de dichas capas proporciona al sistema niveles de abstracción que resuelven muchos problemas de desarrollo, proporcionando reutilización, optimización y refinamiento. Este estilo asegura que los componentes se estructuren en niveles o capas, donde cada componente solo puede interactuar con componentes de su misma capa o bien con otros de capas inferiores. Esto ayuda a reducir las dependencias entre componentes de diferentes niveles. Normalmente hay dos aproximaciones al diseño en capas: Estricto y Laxo.

Un “**diseño en Capas estricto**” limita a los componentes de una capa a comunicarse solo con los componentes de su misma capa o con la capa inmediatamente inferior, el cuál es el que se pone de manifiesto en este trabajo. Un “**diseño en Capas laxo**” permite que los componentes de una capa interactúen con cualquier otra capa de nivel inferior. En la solución arquitectónica del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo se definen tres capas, las cuales dividen la aplicación en tres capas lógicas distintas (18):



**Figura 7.** Arquitectura en tres capas

- ✓ Capa de Presentación (UI): Esta capa se encarga de gestionar todos aquellos aspectos relacionados con la lógica de presentación de la aplicación, como son las validaciones de la información y la gestión de eventos, ya sea del teclado o del mouse. Se comunica con la capa de Negocio.
- ✓ Capa de Lógica del Negocio o Dominio: Se encarga de recibir las peticiones de los usuarios y contiene todo el código que define las reglas de negocio para dar respuestas a las mismas luego de su proceso. Se puede definir como el conjunto de reglas de negocio que abstraen el problema real a tratar.
- ✓ Capa de Acceso a Datos: Contiene los datos necesarios para la aplicación y también puede ofrecer servicios relacionados con la persistencia o recuperación de información, en este caso del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo se encargará de toda la gestión de ficheros y proyectos. En dichos ficheros se realizarán las salvadas de las informaciones necesarias para la construcción y visualización de las columnas litológicas.

### **Ventajas (18)**

- ✓ Reutilización y desarrollos paralelos en cada capa.
- ✓ Facilita la estandarización.
- ✓ Soporta un diseño basado en niveles de abstracción crecientes.
- ✓ Contención de cambios a una o pocas capas.
- ✓ Flexibilidad y alta escalabilidad.

### **Desventajas (18)**

- ✓ A veces no se logra la contención del cambio y se requiere una cascada de cambios en varias capas.

- ✓ Dificultad de diseñar correctamente la granularidad de las capas.

### 2.4 Framework: QT

Qt es un framework<sup>5</sup> de desarrollo para aplicaciones multiplataformas que simplifica mucho el desarrollo de aplicaciones en C++ de forma nativa, también puede ser utilizado en otros lenguajes y funciona en las plataformas Windows y Linux. Es una biblioteca para desarrollar interfaces gráficas de usuario (GUI) y también para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores.

Este framework está compuesto por bibliotecas Qt (clases en C++) y por el diseñador QT Designer, el cual es una potente herramienta para el desarrollo de formularios y presentaciones gráficas para las aplicaciones. Permite acelerar el desarrollo de interfaces de alto rendimiento, a la vez que proporciona una forma fácil de diseñar interfaces gráficas de usuario avanzadas generando el código fuente para las mismas, lo que permite al desarrollador ajustarlo a sus necesidades. El QT Designer utiliza como base la biblioteca gráfica de QT, que ha sido transportada a diversas plataformas, lo que permite que el código generado por el QT Designer pueda ser utilizado en diversas plataformas. Además funciona sólo o asociándose a algunos entornos de desarrollo integrado como Visual Studio .Net o Eclipse. Esta herramienta provee características muy poderosas como la pre-visualización de la interfaz, soporte para widgets y un editor de propiedades bastante poderoso.

Presenta un acceso rápido a la documentación a través de Qt Assistant, donde se puede tener acceso a diferentes ejemplos con sus respectivos códigos fuentes, lo que ayuda a un mejor aprendizaje. La Interfaz de Programación de Aplicaciones(API) de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML; cuenta con una API multiplataforma unificada para la manipulación de archivos y otras para el manejo de ficheros, además de estructuras de datos tradicionales.

#### Otras características:

- ✓ Se distribuye bajo una licencia libre GPL (o QPL), además de la LGPL, que nos permite incorporar las QT en nuestras aplicaciones open-source.
- ✓ Se encuentra disponible para una gran número de plataformas: Linux, MacOS X, Solaris, HP-UX, UNIX con X11, Windows.
- ✓ Es orientado a objetos, lo que facilita el desarrollo de software. El lenguaje para el que se encuentra disponible es C++ aunque han aparecido versiones (conocidas como bindings) para C, Python (PyQt), Java (Qt Jambi), Perl (PerlQt), entre otros.

---

<sup>5</sup> Framework: en el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

- ✓ Es una biblioteca que se basa en los conceptos de widgets (objetos), Señales-Slots y Eventos (Ej.: clic del ratón).
- ✓ Las señales y los slots es el mecanismo para que unos widgets se comuniquen con otros.
- ✓ Los widgets pueden contener cualquier número de hijos. El widget superior (también conocido como top-level), puede ser cualquiera, sea ventana, botón, etc.
- ✓ Compatibilidad multiplataforma con un sólo código fuente.
- ✓ Fácil de internacionalizar.
- ✓ Arquitectura lista para plugins.

Con todo lo anterior expuesto se puede decir que Qt dispone de tres grandes ventajas ante otras bibliotecas:

- ✓ Qt es completamente gratuito para aplicaciones de código abierto, y una de las licencias mencionadas anteriormente, la LGPL, permite que se utilice gratuitamente con fines comerciales.
- ✓ Las herramientas, bibliotecas y clases están disponibles para casi todas las plataformas Unix y sus derivados (como Linux, MacOS X, Solaris, etc.), así como para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquiera de estas plataformas sin necesidad de cambiar el código.
- ✓ Qt tiene una extensa biblioteca con clases y herramientas para la creación de aplicaciones con interfaz de usuario enriquecidas. Estas bibliotecas y clases están bien documentadas, son muy fáciles de usar y tienen una gran herencia de programación orientada a objetos, lo cual hace de la programación de interfaces gráficas un trabajo más ameno.

Este framework es muy potente a la hora de crear interfaces gráficas de usuario, provee componentes que facilitan la elaboración de gráficas, y permiten mostrar informaciones u otros objetos dentro de ellos. Brinda una amplia documentación. Estas son las principales características que dieron paso a la selección del mismo para la realización del sistema que se desarrolla con este trabajo.

### 2.4.1 Biblioteca Qwt

La biblioteca Qwt contiene componentes Qt GUI y clases de utilidad que son principalmente utilizadas por programas con un trasfondo técnico. Proporciona escalas, deslizadores, diales, brújulas, termómetros, ruedas y botones, para controlar o mostrar valores, listas o rangos de tipo doble. Está basada en la biblioteca Qt, por lo que aprovecha todas las características de esta, como son el sistema de señales y slots.

Qwt se distribuye bajo los términos de la Licencia de Qwt, versión 1.0. Además esta biblioteca no distribuye paquetes binarios, pero en la actualidad todos los principales distribuidores de Linux ofrecen uno. Hay que tener en cuenta que estos paquetes a menudo no incluyen los ejemplos. Existen bindings o versiones, para Python (PyQwt) y para Ruby (Korundum / QtRuby). Ejemplos de aplicaciones desarrolladas con Qwt son LabView y TestPoint (19).

El uso de esta biblioteca ahorra una enorme cantidad de tiempo y esfuerzo, pues de lo contrario se tendrían que desarrollar algunos componentes desde el inicio. Solo se debe adaptar los widgets que proporciona a los requisitos funcionales que deben poseer los del editor que se desea desarrollar. De los widgets que brinda esta biblioteca fueron utilizados los que se describen a continuación. El QwtPlot para graficar los datos que se representan mediante curvas, las cuales son configuradas mediante el QwtPlotCurve. El QwtPlotGrid para cuadrricular la columna que contiene las curvas. El QwtPlotPicker para mostrar el valor de la profundidad del pozo que le corresponde a la posición del mouse, cuando se presione el clic izquierdo dentro de la columna de curvas. El QwtScaleWidget para conformar la escala que debe ser graficada en la columna de profundidad.

## 2.6 Lenguaje de Programación: C++

El lenguaje de programación escogido fue C++, el cual abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Además C++ brinda muchas facilidades para la programación orientada a objetos y para el uso de plantillas o programación genérica. Está considerado como un potente lenguaje, ya que trabaja tanto a alto como a bajo nivel, por lo que el código resultante de su compilación es muy eficiente. Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel como son la identificación de tipos en tiempo de ejecución (RTTI) y la posibilidad de redefinir operadores.

La posibilidad de orientar la programación a objetos que nos ofrece este lenguaje permite al programador diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además permite la reutilización del código de una manera más lógica y productiva. Otra importante característica de C++ es la programación modular, ya que una entidad de aplicación en C++ puede estar hecha con varios ficheros de código fuente que son compilados por separado y después unidos. Esta característica permite unir código en C++ con código producido en otros lenguajes.

### Ventajas

Las principales ventajas que presenta el lenguaje C++ son:

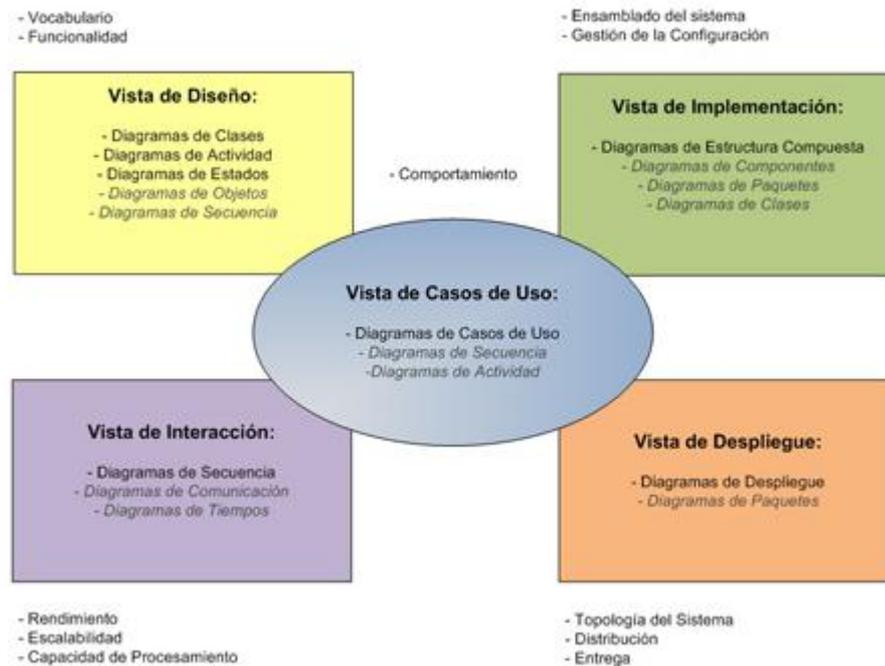
- ✓ Difusión: al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos, páginas web, dedicados a él.

- ✓ Versatilidad: es un lenguaje de propósito general, por lo que se puede emplear para cualquier tipo de problema.
- ✓ Portabilidad: el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- ✓ Eficiencia; es uno de los lenguajes más rápidos en cuanto a ejecución.
- ✓ Herramientas: existen una gran cantidad de compiladores, depuradores y bibliotecas de clases.

Debido a que el lenguaje que utiliza el framework definido (QT) y también la biblioteca seleccionada (QWT) es C++, se selecciona el mismo como el lenguaje para implementar el sistema. También se seleccionó por las ventajas que nos proporciona la programación orientada a objetos descritas anteriormente, así como por su uso en varias plataformas y la rapidez de ejecución, características que están muy presentes tanto en el agrado para el cliente como para un buen funcionamiento del software.

### **2.8 Lenguaje de Modelado: UML**

El Lenguaje Unificado de Modelado (UML) es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de sistemas de software, que da la posibilidad a los desarrolladores de visualizar los resultados de su trabajo en esquemas o diagramas estandarizados, proporcionándoles un vocabulario que incluye tres categorías: elementos, relaciones y diagramas. También, UML suministra mecanismos de extensibilidad y permite la modelación de sistemas con tecnología orientada a objetos. Es un lenguaje gráfico con sintaxis y semántica bien definidas. Este lenguaje reúne las mejores prácticas de ingeniería que han sido probadas con éxito en el modelado de sistemas grandes y complejos. La representación en UML de un software está formada por las 4+1 vistas de la arquitectura (Vista de casos de uso, Vista lógica, Vista de procesos, Vista de implementación, Vista de despliegue), relacionadas entre sí:



**Figura 8** 4+1 Vistas Arquitectónicas

### Ventajas

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas para modelar sistemas, los cuales se mencionan a continuación:

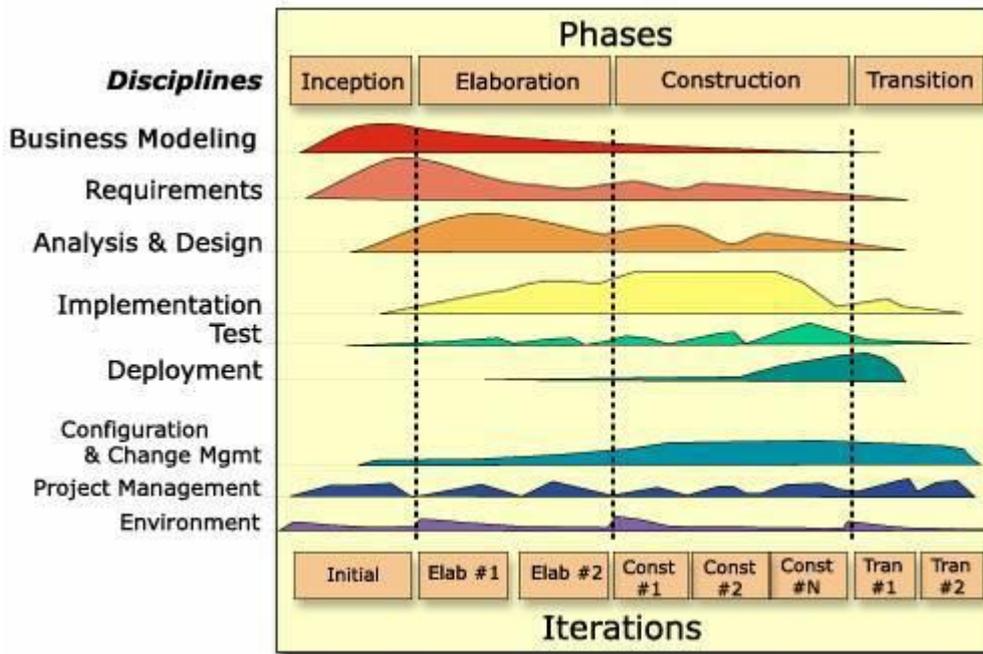
- ✓ Diagramas de casos de uso para modelar los procesos del negocio.
- ✓ Diagramas de secuencia para modelar el paso de mensajes entre objetos.
- ✓ Diagrama de colaboración para modelar interacciones entre objetos.
- ✓ Diagramas de estado para modelar el comportamiento de los objetos en el sistema.
- ✓ Diagramas de actividades para modelar el comportamiento de los casos de uso, objetos u operaciones.
- ✓ Diagramas de clases para modelar la estructura estática de las clases en el sistema.
- ✓ Diagramas de objetos para modelar la estructura estática de los objetos en el sistema.
- ✓ Diagramas de componentes para modelar componentes.
- ✓ Diagramas de implementación para modelar la distribución del sistema.

UML es la consolidación de muchas de las notaciones y conceptos más usados orientados a objetos (20).

## 2.9 Metodología de Desarrollo: Proceso Unificado de Desarrollo (RUP)

Las metodologías de desarrollo surgen por la necesidad de evitar problemas que se presentan en la producción de un software por no seguir normas específicas. Son un conjunto de procedimientos, técnicas y ayudas para el desarrollo de productos software. Cubren todo el ciclo de desarrollo del producto, estableciendo etapas y controles a aplicar en cada momento. Estas recopilan un conjunto de técnicas y procedimientos en cada una de las fases que las componen (21).

El Proceso Unificado de Desarrollo (RUP) es una metodología de desarrollo para proyectos grandes, lo que no quiere decir que no sea también usada en proyectos pequeños, por esto se le denomina una metodología pesada. El ciclo de vida de RUP se divide en cuatro fases de desarrollo (Inicio o Concepción, Elaboración, Construcción, Transición).



**Figura 9.** Ciclo de vida de RUP

Cada una de estas fases contiene sus respectivos hitos y están compuestas por iteraciones las cuales se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada, mostrando como modelar software visualmente capturando la estructura de comportamientos de arquitecturas y componentes.

Esta metodología utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema de software. RUP se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y ser iterativo e incremental. Realiza un levantamiento exhaustivo de requisitos, busca detectar defectos en las fases iniciales con el fin de realizar un proyecto eficiente con calidad y rentable. Intenta reducir el número de cambios que ocurran en el

transcurso del software y realiza el análisis y diseño, tan completo como sea posible con un diseño genérico anticipándose a futuras necesidades. El cliente interactúa con el equipo de desarrollo mediante reuniones.

### **Ventajas**

- ✓ Mitigación temprana de posibles riesgos.
- ✓ Progreso visible desde las primeras etapas
- ✓ Temprana retroalimentación que se ajusta a las necesidades reales.
- ✓ Gestión de la complejidad.
- ✓ El conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.

RUP en su modelado simplifica la realidad, proporciona los planos del sistema e incluye elementos que tienen gran influencia, omitiendo aquellos menores que no son relevantes para el nivel de abstracción dado. Los modelos ayudan a visualizar como es o como se quiere que sea el sistema, permiten especificar la estructura o comportamiento del sistema proporcionando plantillas que guiarán y documentarán las decisiones que se adopten. Lo planteado anteriormente da paso a la selección de esta metodología de desarrollo.

## **2.10 Herramienta Case: Visual Paradigm**

Visual Paradigm (VP) es la herramienta CASE escogida para el desarrollo de esta arquitectura de software. Se seleccionó esta herramienta debido a que posee licencia gratuita y comercial, es un producto de mucha calidad, configurable para el uso en varios idiomas, muy fácil de instalar y actualizar además de brindar compatibilidad entre ediciones. Esta herramienta soporta modelado UML y provee el modelado de procesos de negocios.

Visual Paradigm UML responde rápidamente y con poca memoria, permitiendo manejar grandes y complicadas estructuras de un proyecto de forma eficiente y que solo requiera de una configuración de escritorio. Visual Paradigm presenta varias ediciones, cada una destinada a sus propias necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. También ayuda a los equipos de desarrollo de software a desplegar el proceso de desarrollo de los mismos, logrando maximizar y acelerar tanto las contribuciones individuales como las de equipo, facilita la diagramación visual y el diseño de sus proyectos, además posee alta interoperabilidad.

### **Ventajas:**

- ✓ Navegación intuitiva entre código y el modelo.
- ✓ Poderoso generador de documentación y reportes UML PDF/HTML/MS Word.
- ✓ Demanda en tiempo real, modelo incremental de viaje redondo y sincronización de código fuente.

- ✓ Superior entorno de modelado visual.
- ✓ Soporte completo de notaciones UML.
- ✓ Diagramas de diseño automático sofisticado.
- ✓ Análisis de texto y soporte de tarjeta CRC.
- ✓ Lenguajes en generación de Código e Ingeniería Inversa.
- ✓ Creación de modelos UML.
- ✓ Modelado de base de datos.
- ✓ Mapa de relación de objetos.
- ✓ Integración IDE.

### **2.11 Conclusiones Parciales**

Luego de analizadas las principales tecnologías que serán utilizadas en la elaboración del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo se arriban a las siguientes conclusiones:

El uso del Framework Qt reducirá el costo de tiempo y esfuerzo para la implementación del Sistema, así como el uso de un estilo y estándar de codificación permitirá un mejor entendimiento del código para próximas versiones del sistema.

Las herramientas y tecnologías de software libre garantizan el desarrollo de un software con la calidad, el tiempo y el costo requerido; además de que propician una mejor aceptación del software por parte del cliente.

### Capítulo 3: Construcción y validación de la solución propuesta

#### 3.1 Introducción

En este capítulo se describen todos los aspectos relacionados con la construcción del sistema, así como la validación del resultado. Se analizan las características del sistema, incluyendo los requisitos, modelo de diseño y el modelo de casos de uso del sistema. Se realiza una descripción del modelo de diseño debido a los cambios que han surgido desde la etapa de análisis y diseño hasta la implementación del mismo. También se describe la vista lógica y de implementación con sus respectivos diagramas y finalmente se especifican las pruebas de caja negra realizadas al sistema.

#### 3.2 Características del Sistema

En el sistema que se desarrolla todo el proceso comienza cuando es entregado al Especialista en Sedimentología el Reporte del pozo y concluye con la creación y envío de la Columna Litológica del pozo en cuestión al Geólogo del pozo. El sistema debe permitir graficar las columnas curvas representando los gases, porosidad y otros componentes que necesitan graficarse mediante curvas. También debe permitir construir las columnas litológicas de los pozos de petróleo. El especialista en sedimentología puede dejar plasmado descripciones de las distintas litologías que no podrán expresarse mediante la columna gráfica.

##### 3.2.1 Requisitos Funcionales

“Son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares” (22). Describen las funcionalidades o servicios que se espera que posea el software o sistema y permiten mantener un acuerdo con el cliente al respecto. Muchos de los problemas que enfrenta actualmente la Ingeniería de software están dados precisamente por errores o imprecisiones en el momento de especificar los requisitos. Es por ello que algo de vital importancia al respecto es la consistencia y completación de los mismos (2).

Después de realizar un análisis detallado del Negocio se logró identificar como Requisitos Funcionales del sistema los siguientes (2):

**RF 1.** Gestionar proyecto.

**RF 1.1** El sistema debe permitir crear un nuevo proyecto.

**RF 1.2** El sistema debe permitir abrir un proyecto existente.

**RF 1.3** El sistema debe permitir guardar los cambios producidos sobre un proyecto existente.

**RF 1.4** El sistema debe permitir eliminar un proyecto existente.

**RF 2.** El sistema debe permitir exportar ficheros.

**RF 2.1** El sistema debe permitir exportar los datos referentes a las litologías en las extensiones .pdf y .xml.

**RF 2.2** El sistema debe permitir exportar los datos referentes a las curvas en las extensiones .pdf y .xml.

**RF 2.3** El sistema debe permitir exportar los datos referentes a las descripciones en las extensiones .pdf y .xml.

**RF 3.** El sistema debe permitir importar ficheros.

**RF 3.1** El sistema debe permitir importar los datos referentes a las curvas en las extensiones .las y .xml.

**RF3 .2** El sistema debe permitir importar las descripciones en las extensiones .las y .xml.

**RF 4.** El sistema debe permitir construir la columna Curvas.

**RF 4.1** El sistema debe permitir graficar la curva que representa la Velocidad de perforación.

**RF 4.2** El sistema debe permitir graficar la curva que representa los Gases.

**RF4.3** El sistema debe permitir graficar la curva que representa la Calcimetría.

**RF 5.** El sistema debe permitir construir la columna Litologías.

**RF 5.1** El sistema debe permitir insertar en la columna Litologías los símbolos clasificados como Litologías, en correspondencia con los datos obtenidos del análisis de la muestra en un intervalo dado.

**RF 5.2** El sistema debe permitir insertar en la columna Litologías los símbolos clasificados como Fósiles sobrescribiendo el símbolo Litología, en correspondencia con los datos obtenidos del análisis de la muestra en un intervalo dado.

**RF 5.3** El sistema debe permitir insertar en la columna Litologías los símbolos clasificados como Texturas sobrescribiendo el símbolo Litología, en correspondencia con los datos obtenidos del análisis de la muestra en un intervalo dado.

**RF 5.4** El sistema debe permitir insertar en la columna Litologías los símbolos clasificados como Minerales sobrescribiendo el símbolo Litología, en correspondencia con los datos obtenidos del análisis de la muestra en un intervalo dado.

**RF 5.5** El sistema debe permitir eliminar un símbolo.

**RF 6.** El sistema debe permitir construir la columna Descripciones.

**RF 6.1** El sistema debe proveer un campo de texto en la columna Descripción para insertar una breve descripción de la litología en la sección que se esté analizando.

**RF 6.2** El sistema debe permitir realizar cambios en una descripción existente.

**RF 6.3** El sistema debe permitir eliminar una descripción.

### 3.2.2 Requisitos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los mismos están vinculados con los requisitos funcionales, es decir, al conocer lo que el sistema debe hacer, se puede determinar cómo ha de comportarse. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto además de formar una parte significativa de las especificaciones. También ayudan a marcar la diferencia entre un producto bien aceptado por los usuarios y uno con poca aceptación. Para el desarrollo de la actual aplicación se tuvo en cuenta los siguientes requisitos no funcionales (18):

#### 3.2.2.1 Usabilidad

- ✓ La aplicación debe ser de fácil comprensión, navegación, configuración y utilización tanto para usuarios con un nivel alto de experiencia, como de niveles inferiores en el campo de la informática.
- ✓ La información debe ser mostrada de forma lógica y organizada.

#### 3.2.2.2 Apariencia o Interfaz Externa

- ✓ El sistema debe brindar una interfaz externa de diseño sencillo y accesible a cualquier tipo de usuario, independientemente de su nivel de conocimiento de la aplicación.

### 3.2.2.3 Rendimiento

- ✓ El sistema debe tener un eficiente tiempo de respuesta, de acuerdo con la funcionalidad que se esté usando en cualquier momento, no excediendo los tres segundos para las funcionalidades que no sean muy complejas.

### 3.2.2.4 Requerimiento de ayuda y documentación

- ✓ El sistema debe contar con un manual de usuario.
- ✓ El sistema debe contar con una ayuda disponible en cualquier momento que el usuario desee realizar una consulta para facilitar su trabajo con el mismo.

### 3.2.2.5 Requisitos de Hardware

- ✓ Se requiere una PC Pentium III o superior.
- ✓ Se requiere 512 MB de RAM o superior.

### 3.2.2.6 Portabilidad

- ✓ El sistema debe ser compatible con los sistemas operativos Windows y Linux.

## 3.2.3 Modelo de Casos de Uso del Sistema

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema (23).

### 3.2.3.1 Actores del Sistema

Un actor es un usuario del sistema. Incluye usuarios humanos y otros sistemas computarizados. Un actor usa un caso de uso para desempeñar alguna porción de trabajo que es de valor para el negocio. El conjunto de casos de uso al que un actor tiene acceso define su rol global en el sistema y el alcance de su acción (23). A continuación se realiza una descripción de los actores que van a interactuar con el sistema que se desarrolla.

**Tabla 1** Descripción de los actores del sistema

Descripción de los actores del sistema	
Actores	Descripción
Especialista en Sedimentología	Es la persona encargada de realizar las operaciones necesarias para graficar la

	Columna Litológica.
Usuario	Es la persona familiar al negocio que puede realizar algunas de las actividades contempladas dentro de la funcionalidad del software, pero tiene restringidas la mayoría.

### 3.2.3.2 Casos de Uso del Sistema

Un Caso de Uso es una unidad simple de trabajo significativo; por ejemplo, "Validarse en el sistema", "Registrarse en el sistema" y "Crear un pedido" son todos casos de uso (23). Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso o "extender" a otro caso de uso con su propio comportamiento (23).

Para el desarrollo del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo se definieron 20 requisitos que fueron agrupados en 11 Casos de Uso del sistema de acuerdo a su impacto en la arquitectura, de la siguiente manera (18):

**Tabla 2** Clasificación de los Casos de Uso del Sistema

Casos de Uso del Sistema			
Críticos	Secundarios	Auxiliares	Opcionales
Importar Ficheros	Gestionar Usuario	Exportar Ficheros	Imprimir proyecto
Construir Columna Curvas	Autenticar Usuario	Crear Litología	Modificar Contraseña
Construir Columna Litologías			
Construir Columna Descripciones			
Gestionar Proyecto			

Los casos de uso identificados se pueden clasificar en críticos, secundarios, auxiliares y opcionales según su prioridad para la arquitectura (18).

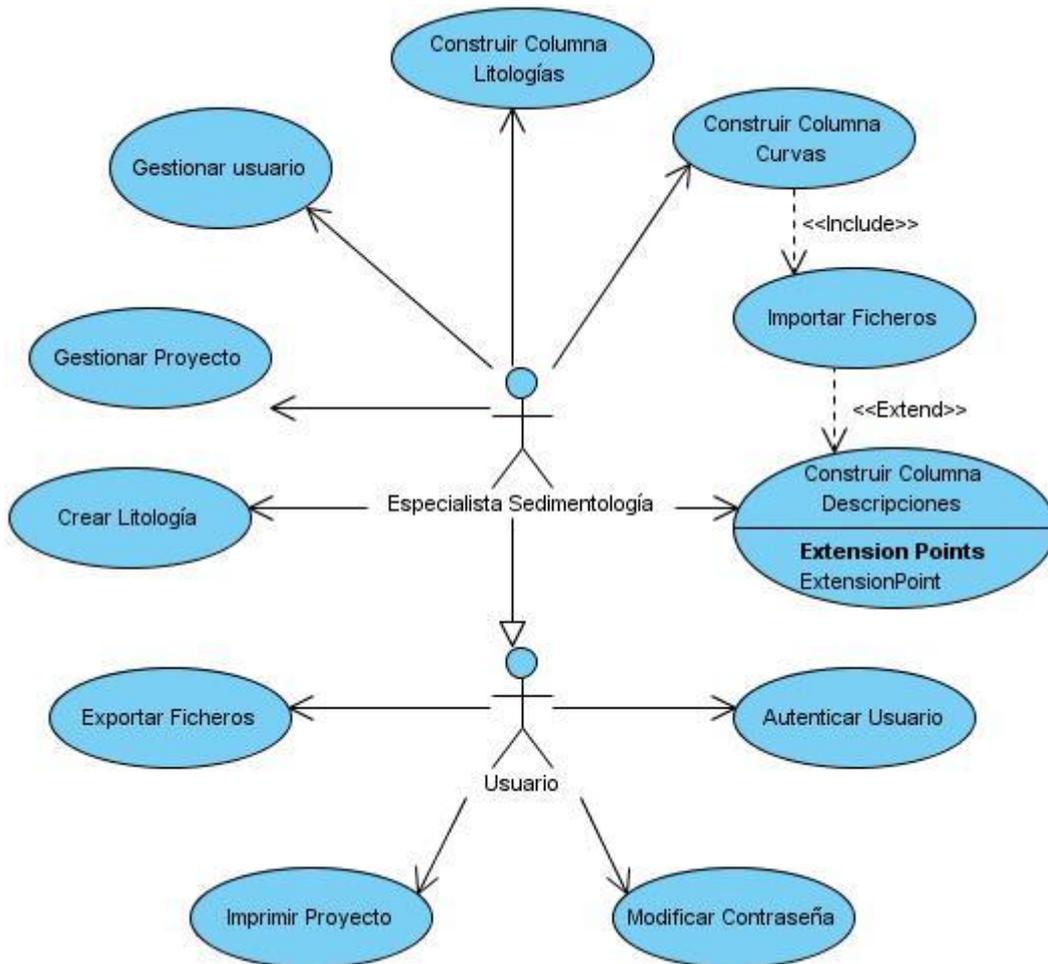
- ✓ Los casos de uso críticos contienen las principales funcionalidades que el sistema debe cumplir y que son de vital importancia para los usuarios, por lo que definen la arquitectura básica.
- ✓ Los casos de uso secundarios sirven de apoyo a los casos de uso críticos, contienen aquellas funcionalidades de menor impacto sobre la arquitectura, aunque también deben implementarse tempranamente pues responden a requisitos de interés para los usuarios.
- ✓ Los casos de uso auxiliares no son claves para la arquitectura y son usados sobre todo para completar casos de uso críticos o secundarios.
- ✓ Los casos de uso opcionales responden a funcionalidades cuya existencia es optativa para la aplicación y no son imprescindibles en las primeras versiones.

Otra clasificación importante para los casos de uso, son aquellos que se declaran arquitectónicamente significativos, ya que describen funcionalidades imprescindibles para el sistema a través de las cuales se valida la arquitectura propuesta para el mismo. Para el sistema se clasificaron 5 casos de uso arquitectónicamente significativos atendiendo a los criterios mencionados anteriormente (18):

- ✓ Importar Ficheros
- ✓ Construir Columna Curvas
- ✓ Construir Columna Litologías
- ✓ Construir Columna Descripciones
- ✓ Gestionar Proyecto

En el sistema que se desarrolla se implementan los casos de uso arquitectónicamente significativos y el caso de uso auxiliar "Exportar Ficheros", debido a que son las funcionalidades más importantes del sistema. Con la correcta implementación de estas funcionalidades el sistema tendrá un buen funcionamiento, por lo que se podrá ir utilizando mientras se le da cumplimiento al resto de las funcionalidades que se incorporarán en versiones futuras del software.

Para un mejor entendimiento de la distribución de los casos de uso definidos, actores del sistema y las relaciones entre ellos se presenta a continuación el Diagrama de Casos de Uso del Sistema:



**Figura 10** Diagrama de Casos de Uso del Sistema

### 3.3 Modelo de Diseño

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida. En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además, impone una estructura del sistema que se debe conservar lo más fielmente posible cuando se esté desarrollando el sistema (24).

#### 3.3.1 Diagrama de Clases del Diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. El diseño se utiliza para que el programador en cuestión pueda realizar una correcta implementación del sistema a desarrollar (24). El diagrama incluye la siguiente información (25):

- ✓ Clases, asociaciones y atributos.
- ✓ Interfaces, con sus operaciones y constantes.
- ✓ Métodos.
- ✓ Navegabilidad.
- ✓ Dependencias.

Un Diagrama de Clases de Diseño muestra definiciones de entidades de software, más que conceptos del mundo real. El diagrama de clases del diseño es el diagrama más importante de la fase de diseño de un software, este brinda una visión general para comenzar con la implementación del producto. Para la creación del diagrama de clases del diseño primeramente se identifican las entidades de software necesarias para desarrollar el sistema que se pretende realizar, se definen las relaciones existentes entre ellas y los atributos y métodos según la necesidad del sistema. Para la definición de forma más sencilla de los métodos en cada clase es muy factible el apoyo en los diagramas de interacción (colaboración o secuencia).

Es importante destacar que en el sistema que se desarrolla al igual que en la mayoría de los software el diagrama de clases del diseño se actualiza, pues aparecen nuevos elementos, a medida que avanza el desarrollo del sistema hasta obtener la versión final que se corresponde con el diseño definitivo del software, por lo cual se hace necesario incluir en este documento el diagrama de clases del diseño correspondiente al software que se desarrolla, el cual se muestra a continuación.

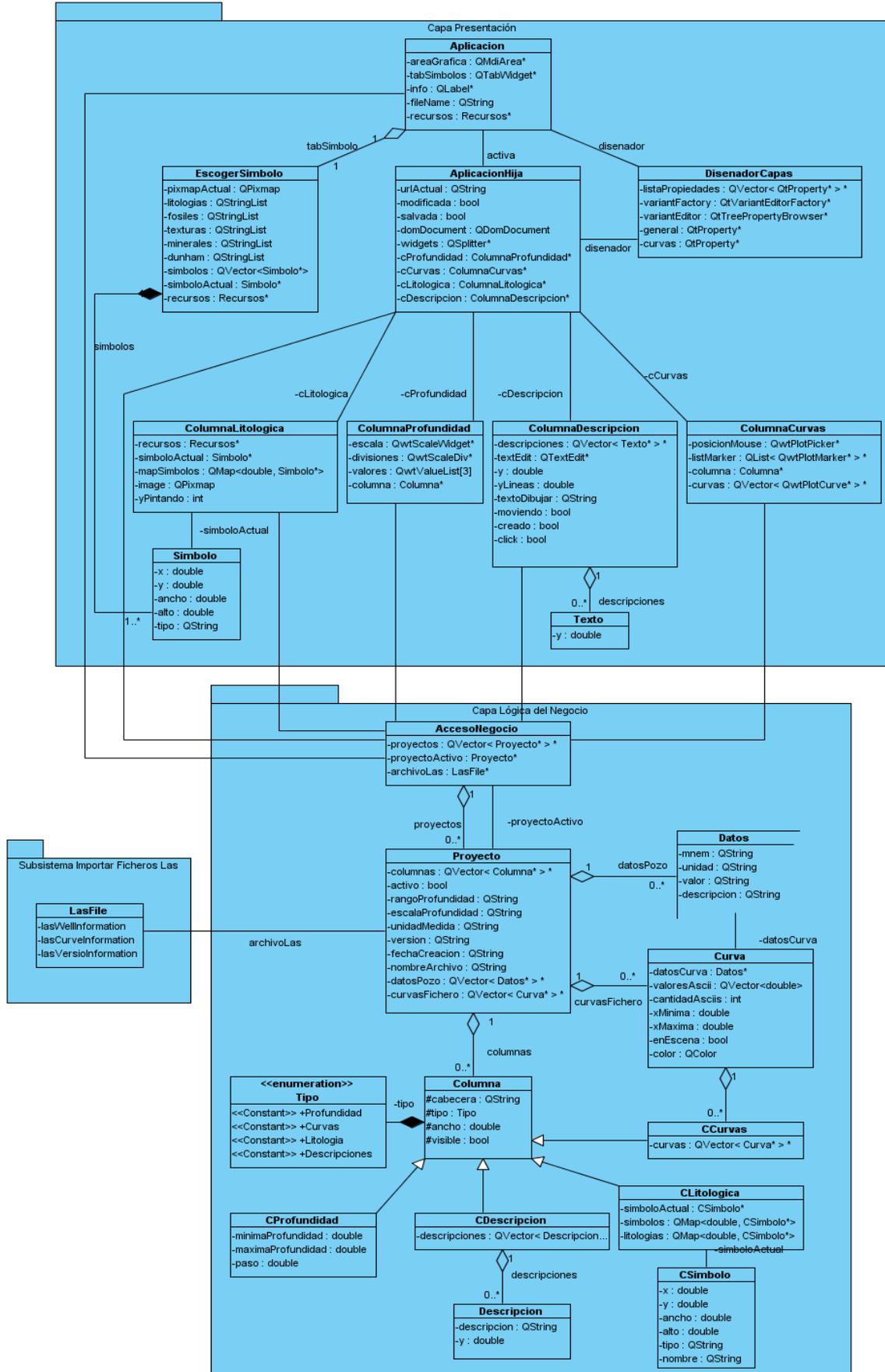


Figura 11 Diagrama de clases del diseño

### 3.3.2 Patrones de diseño utilizados

En el desarrollo del sistema se utilizan varios patrones arquitectónicos que definen la estructura del software a desarrollar, los cuales tienen una serie de directivas para organizar los componentes con el objetivo de facilitar la tarea del diseño del sistema. Estos patrones garantizarán una mayor comprensión de la arquitectura del sistema y agilizarán el posterior desarrollo e implementación del mismo (18).

#### 3.3.2.1 Patrones GoF

**Facade (Fachada):** Propone la selección de un objeto que represente un sistema u organización global para que sea controlador. Las fachadas se suelen utilizar para proporcionar un "frontal de servicios" de la capa de negocio. De este modo, la interacción de los clientes con esta capa se simplifica considerablemente (26). Mediante este patrón se selecciona la clase AccesoNegocio como la adecuada para que se encargue de la interacción con las demás clases del sistema.

**Singleton:** Se utiliza este patrón ya que garantiza que exista una instancia única para una clase y proporciona un punto de acceso global a ella (18). El patrón Singleton está presente en la clase AccesoNegocio, la cual posee una estructura de forma tal que no se puede crear más de una instancia de la misma. También posee un método llamado objeto, el cual proporciona un acceso global a esta clase.

#### 3.3.2.2 Patrones GRASP

En el patrón **Experto** la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada (18). Este patrón fue aplicado en las clases de la capa de lógica del negocio que contienen los datos necesarios para dar respuesta a las funcionalidades definidas en cada una de ellas.

**Bajo Acoplamiento.** Proporciona que entre las clases exista la menor cantidad de dependencias posibles. De tal forma que en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Este patrón es respetado durante la implementación del sistema pues se ha incluido la menor cantidad posible de dependencias entre las clases.

La **Alta Cohesión** consiste en que cada elemento de un diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo (18). En el sistema que se desarrolla, existe colaboración entre las clases para realizar tareas con cierta dificultad. Por ejemplo, si para realizar una funcionalidad se necesita llevar a cabo varias labores, estas son asignadas a las clases responsables de realizarlas.

El patrón **Controlador** asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas, facilitando la centralización de actividades (validaciones y seguridad) (18). Este patrón está presente en la clase AccesoNegocio, la cual controla todo el flujo de las funcionalidades del sistema y da respuesta a las mismas asignándoles tareas a las demás clases con las que mantiene una alta cohesión.

### 3.4 Estándar de Codificación

Esta es una visión general de la convención de codificación que se usa al escribir el código del Sistema para Graficar Columnas Litológicas. Los datos han sido elaborados teniendo en cuenta el estándar de codificación que se utiliza en el framework Qt y mediante el intercambio entre varios desarrolladores.

#### 3.4.1 Declaración de variables

- ✓ Declarar cada variable en una línea separada, de esta forma cada variable se puede comentar por separado.
- ✓ Evitar los nombres cortos siempre que sea posible (por ejemplo, "a ", "na"). Solo están bien utilizados en contadores, temporales u otros usos donde el uso de la variable es obvio.

```
// Correcto
int alto;
int ancho;

// Incorrecto
int a, b;
```

**Figura 12** Declaración de variables. Nombres cortos

- ✓ No declarar una variable hasta que sea necesitada.
- ✓ Las variables y funciones deben comenzar con una letra minúscula, el resto de las palabras empiezan con una letra mayúscula. No se utilizará en ningún caso el caracter "\_".
- ✓ Evitar las abreviaturas.

```
// Correcto
short control;

// Incorrecto
short ctrl;
```

**Figura 13** Declaración de variables. Abreviaturas

- ✓ Las clases siempre empiezan con una letra mayúscula y el resto de las palabras también deben empezar con mayúscula. No se utilizará en ningún caso el caracter "\_".
- ✓ Las constantes: todo en mayúscula. Si son varias palabras, unidas por el carácter "\_".

### 3.4.2 Espacios en blanco

- ✓ Siempre usar una sola línea en blanco.
- ✓ Siempre utilizar un solo espacio después de una palabra clave y entre los operadores aritméticos.

```
// Correcto
if (a && b)
{

}

// Incorrecto
if(a&& b)
{
}
}
```

**Figura 14** Espacios en blanco. Palabras clave y llaves

- ✓ Para los punteros o referencias, utilizar siempre un espacio entre el '\*' o '&' y el nombre de la variable, pero no un espacio entre el tipo y '\*' o '&'.

```
// Correcto
char *y;
const int &miEntero;

// Incorrecto
char* y;
const int& miEntero;
```

**Figura 15** Espacios en blanco. Punteros o referencias

### 3.4.3 Llaves

- ✓ Las implementaciones de instrucciones, funciones y declaraciones de clases siempre tiene la llave izquierda en el comienzo de una línea:

```
// Correcto
class Persona
{
    static void funcion()
    {

    }
};

// Incorrecto
class Persona{
    static void funcion(){

    }
};
```

**Figura 16** Llaves. Clases y funciones

- ✓ Utilice llaves cuando el cuerpo de una instrucción condicional contiene más de una línea, y también si una única línea es muy compleja.

```
// Correcto
if (vacio)
    return false;

// Incorrecto
if (vacio)
{
    return false;
}
```

**Figura 17** Llaves. Condicionales

- ✓ Excepción: Utilice llaves también si la instrucción primaria abarca varias líneas:

```
// Correcto
if (vacio || !esValido()
|| !codigo)
{
    return false;
}
```

**Figura 18** Llaves. Instrucción de varias líneas

- ✓ Utilice llaves cuando el cuerpo de una sentencia condicional está vacía:

```
// Correcto
while (a) {}

// Incorrecto
while (a);
```

Figura 19 Llaves. Cuerpo vacío

### 3.4.4 Paréntesis

- ✓ Utilice paréntesis para agrupar expresiones:

```
// Correcto
if ((a && b) || c)

// Incorrecto
if (a && b || c)
```

Figura 20 Paréntesis. Agrupar Expresiones

### 3.4.5 Saltos de línea

- ✓ Mantener las líneas con menos de 100 caracteres, insertar saltos si es necesario.
- ✓ Las comas van al final de un salto de línea, los operadores comienzan en el principio de la nueva línea.

```
// Correcto
if (largaExpresion
+ otraLargaExpresion
+ otrOtraLargaExpresion)
{
}

// Incorrecto
if (largaExpresion +
otraLargaExpresion +
otraOtraLargaExpresion)
{
}
```

Figura 21 Paréntesis. Operadores y comas

### 3.4.6 Excepción General

- ✓ Se puede romper una regla siempre y cuando sea para que el código se vea bien.

### 3.5 Implementación

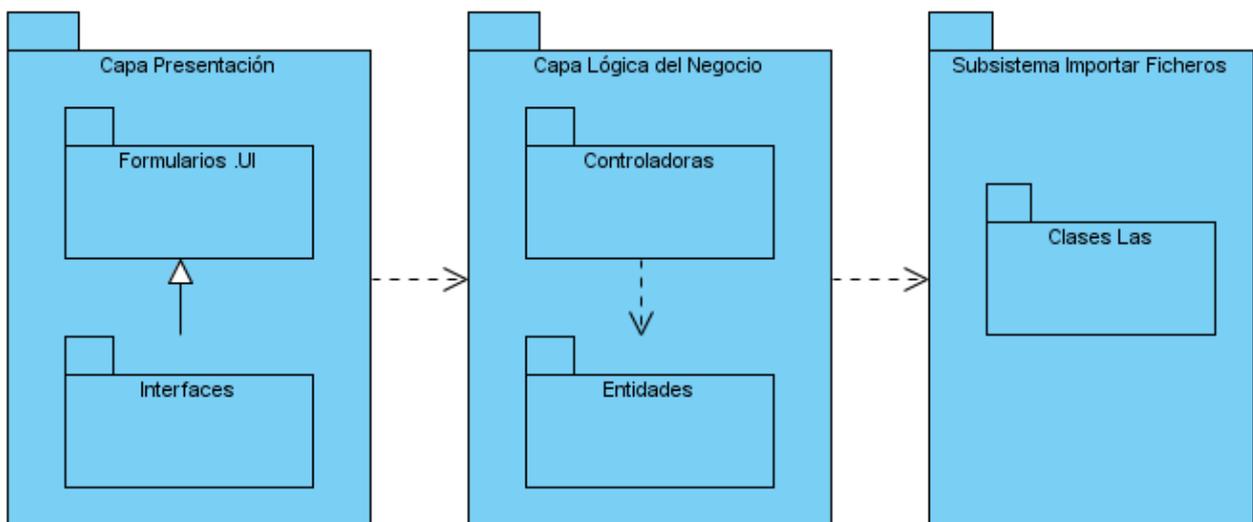
En la implementación se empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares (24).

#### 3.5.1 Modelo de Implementación

El Modelo de Implementación describe como las clases se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. El Modelo de Implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modelación disponible en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes uno de otros (24).

##### 3.5.1.1 Vista Lógica

La vista lógica se basa fundamentalmente en lo que el sistema debe brindar en términos de servicios a sus usuarios. Es una abstracción del modelo de diseño e identifica a gran escala paquetes, subsistemas, clases, y las relaciones que existen entre ellas, ya sean de dependencia o de uso. Esto brinda una descomposición que potencia el análisis funcional, además de servir para identificar mecanismos y elementos de diseño comunes a diversas partes del sistema. El siguiente diagrama muestra la vista lógica del sistema que se desarrolla:



**Figura 22** Vista Lógica del Sistema

En la Capa de Presentación están contenidas todas las clases interfaces, las cuales son las encargadas de visualizar todos los componentes (botones, menús, barras de herramientas, entre

otros) con los que interactúa el usuario, mediante los datos que provee la Capa Lógica de Negocios y las características y funcionalidades que son heredadas de los formularios (.UI).

La Capa Lógica de Negocios contiene las clases controladoras que reciben las peticiones de la capa de presentación y dan respuestas a la misma luego de procesar la información recibida, de realizar los cálculos pertinentes y accediendo a los datos necesarios de las clases entidades. Estas clases describen objetos, reales o abstractos, que existen en la lógica del negocio del sistema, del cual deseamos guardar información.

En el Subsistema de Importación de Ficheros están presentes todas las clases que controlan la gestión de los ficheros con extensión .las, donde se pueden encontrar funcionalidades como crear un fichero, abrirlo y leer toda la información contenida en éste.

### 3.5.1.2 Vista de Implementación

La Vista de Implementación se enfoca en la descomposición del software en capas y subsistemas de implementación, ya sean componentes de código fuente, binarios o ejecutables. Presenta la organización de los módulos del software en el ambiente de desarrollo del mismo; además provee una vista de la trazabilidad de los elementos de diseño de la vista lógica para la implementación. Con esta vista se abren caminos para los desarrolladores y para la visualización de la implementación, permitiendo tomar decisiones respecto a las tareas del desarrollo. A continuación se muestra una vista general de la implementación, la misma ayuda a un mejor entendimiento de cómo se realizó la implementación del sistema.

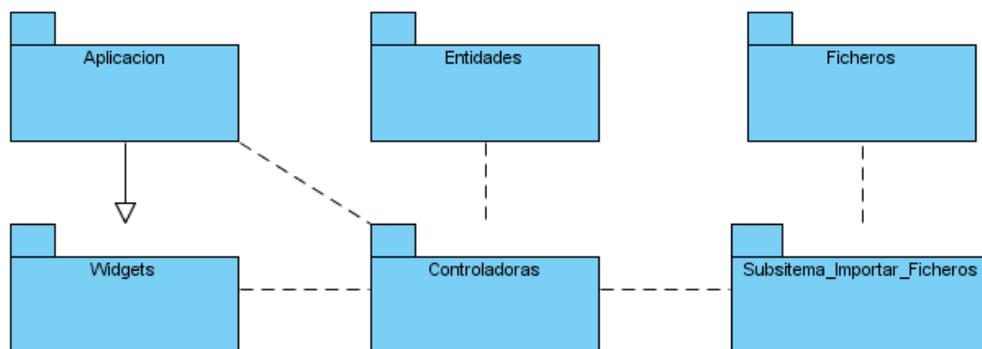


Figura 23 Vista General de Implementación

### 3.5.1.3 Descripción de los componentes de la Vista de Implementación

**Aplicación:** Módulo que contendrá los paquetes de clases o componentes generales de la aplicación.

**Widgets:** Contiene todas las ventanas y componentes (botones, menús, barras de herramientas, editores de texto, etc.) que serán visualizadas en el sistema, en los cuales se muestran los datos que les provea la capa de lógica del negocio.

**Controladoras:** Contiene toda la lógica de negocio del sistema, es decir, algoritmos necesarios y reglas del negocio. Responden a las peticiones realizadas por la aplicación utilizando para esto los datos de las clases entidades y el subsistema de importar ficheros para tener acceso a los ficheros que contienen datos necesarios para el buen funcionamiento del sistema.

**Entidades:** Contienen los datos de los objetos de la lógica del negocio, ya sean abstractos o reales, pero que estén contenidos dentro de la lógica del sistema desarrollado.

**Subsistema\_Importar\_Ficheros:** Contiene los elementos necesarios para la importación de datos que estén almacenados en ficheros con extensión .las.

**Ficheros:** Representa a los ficheros con extensión .las.

### 3.5.1.4 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre sus elementos. Los componentes representados pueden ser datos, archivos, ejecutables, código fuente y directorios (24). A continuación se muestra el correspondiente al sistema que se desarrolla.

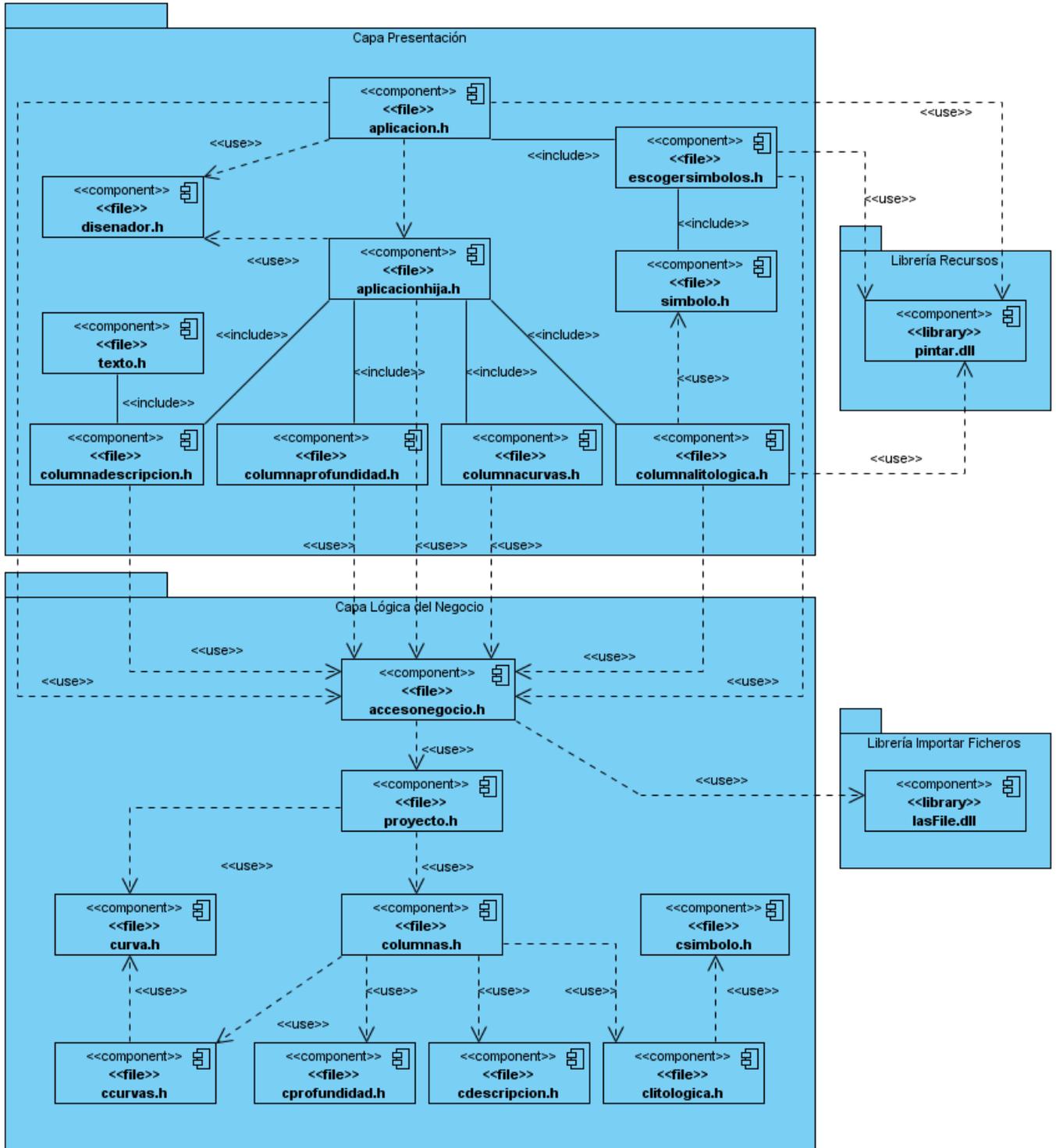


Figura 24 Diagrama de Componentes

### 3.6 Fase de Prueba

Una vez desarrollados los componentes gráficos, básicos y avanzados, es necesario realizarles pruebas, debido a que su ejecución no implica que esté libre de fallas. Es por ello que este epígrafe se

centra en las pruebas. El objetivo principal es validar más detalladamente cada una de las soluciones que se dieron en la implementación.

La fase de prueba en el desarrollo de software consiste en probar las aplicaciones construidas. Las pruebas permiten verificar, validar y revelar la calidad de un software. Gran porcentaje de los programas que se desarrollan tienen errores, en esta fase de pruebas es donde se encuentran y corrigen generalmente los errores de las fases anteriores. Es por ello que probar es una de las fases más importantes para obtener un producto final con la calidad requerida.

### 3.6.2 Pruebas de Caja Negra

Las pruebas de caja negra se concentran en los requisitos funcionales del software; estas técnicas de prueba se concentran en el dominio de la información del software, derivando casos de prueba mediante partición de los dominios de entrada y salida de un programa en forma tal que proporcione cobertura completa (27). El objetivo de realizar este tipo de prueba, es para detectar el incorrecto o incompleto funcionamiento del sistema, comprobar que la entrada se acepta de forma adecuada y que se produce un resultado correcto.

#### 3.6.2.1 Casos de Pruebas

Un caso de prueba permite detallar la forma en que se va a probar el sistema, incluyendo los datos de entrada con las que se realizará la prueba correspondiente, las condiciones de ejecución y resultados obtenidos (24).

Deben verificar:

- ✓ Si el producto satisface los requisitos del usuario, tal y como se describe en las especificación de los requisitos.
- ✓ Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Para verificar que se cumplieran los requisitos funcionales establecidos anteriormente, se le realizaron pruebas a cada una de las secciones contenidas en los distintos casos de uso de este trabajo, conformando una interfaz donde se muestra el resultado esperado.

##### 3.6.2.1.1 Caso de prueba para el caso de uso del sistema “Gestionar Proyecto”

Descripción General: El caso de uso es uno de los más importantes del software, ya que es el encargado del trabajo con los distintos proyectos del sistema.

Condiciones de Ejecución: Acceder al menú “Archivo” o a la barra de herramientas.

**Tabla 3** Caso de Prueba de la sección Nuevo Proyecto del caso de uso “Gestionar Proyecto”.

<b>Sección Crear Nuevo Proyecto.</b>			
Escenarios de la sección	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
EC 1.1 Crear un nuevo proyecto satisfactoriamente.	El sistema permite crear una nueva ventana en el área de trabajo con los datos iniciales correspondientes al nuevo proyecto.	El sistema crea una nueva ventana correspondiente al nuevo proyecto.	<p>El usuario escoge la opción nuevo proyecto.</p> <p>El sistema muestra un formulario para escoger la escala y la unidad de medida con la que se trabajará en el proyecto.</p> <p>El usuario escoge los datos (escala y la unidad de medida).</p> <p>El sistema comprueba que se hayan escogido las dos opciones y crea una nueva ventana en el área de trabajo.</p>
EC 1.2 Crear un nuevo proyecto dejando campos sin escoger.	No se crea el proyecto solicitado.	El sistema muestra un mensaje de error avisando que se deben escoger los dos campos.	<p>El usuario escoge la opción nuevo proyecto.</p> <p>El sistema muestra un formulario para escoger la escala y la unidad de medida con la que se trabajará en el proyecto.</p> <p>El sistema comprueba que se haya escogido una escala y una unidad de medida y luego muestra un mensaje de error avisando que se deben escoger los dos campos.</p>

**Tabla 4** Caso de Prueba de la sección Abrir Proyecto del caso de uso “Gestionar Proyecto”.

<b>Sección Abrir Proyecto.</b>			
Escenarios de la sección	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
EC 1.1: Abrir un proyecto satisfactoriamente.	El sistema permite abrir un proyecto existente.	El sistema abre el fichero correctamente creando una nueva ventana en el área de trabajo con los	<p>El usuario escoge la opción abrir proyecto.</p> <p>El sistema muestra la ventana para seleccionar el fichero que se desea abrir.</p> <p>El usuario escoge el fichero y oprime el</p>

		datos contenidos en el fichero seleccionado.	botón aceptar. El sistema crea una nueva ventana en el área de trabajo con los datos de las columnas (Profundidad, Curvas, Litologías, Descripción) contenidos en el fichero seleccionado.
EC 1.2: Abrir un proyecto con datos incorrectos.	No se abre el fichero el fichero seleccionado.	El sistema no realiza ninguna acción, siendo esto una no conformidad.	El usuario escoge la opción abrir proyecto. El sistema muestra la ventana para seleccionar el fichero que se desea abrir. El usuario escoge el fichero y oprime el botón aceptar.

**Tabla 5** Caso de Prueba de la sección Salvar Proyecto del caso de uso "Gestionar Proyecto".

<b>Sección Salvar Proyecto.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1: Salvar un proyecto correctamente.	El sistema salva el proyecto activo en el directorio seleccionado.	Salva correctamente el proyecto que está activo en ese momento.	El usuario escoge la opción salvar proyecto. El sistema muestra la ventana para la selección del directorio y del nombre del fichero a salvar y seguidamente crea el fichero con todos los datos de las columnas del proyecto activo en el área de trabajo.
EC 1.2: Salvar un proyecto sin proyectos creados o sin modificaciones en un proyecto.	Al no haber proyectos creados, no se salva ningún proyecto.	Muestra un mensaje informando que no hay proyectos creados o no se han efectuados cambios.	El usuario escoge la opción salvar proyecto. El sistema verifica que haya algún proyecto creado y muestra un mensaje informando que no hay proyectos creados o no se han efectuados cambios.

**Tabla 6** Caso de Prueba de la sección Eliminar Proyecto del caso de uso “Gestionar Proyecto”.

<b>Sección Eliminar Proyecto.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1: Eliminar un proyecto satisfactoriamente.	El sistema elimina un fichero correspondiente a un proyecto.	Elimina satisfactoriamente el fichero seleccionado.	<p>El usuario escoge la opción eliminar proyecto.</p> <p>El sistema muestra una ventana donde puede introducir la dirección del fichero a eliminar o puede pulsar en el botón “Aceptar” para escoger la dirección.</p> <p>El usuario escoge la dirección del fichero y pulsa sobre el botón “eliminar”.</p> <p>El sistema elimina el fichero seleccionado.</p>

### 3.6.2.1.2 Caso de prueba para el caso de uso del sistema “Exportar Ficheros”

Descripción General: El caso de uso se inicia cuando el usuario selecciona la opción exportar PDF o Imagen en el menú “Archivo” o en la barra de herramientas en el caso de la opción exportar PDF. Este caso de uso genera los datos de todas las columnas del proyecto activo.

Condiciones de Ejecución: Acceder al menú “Archivo” o a la barra de herramientas y tener algún proyecto creado.

**Tabla 7** Caso de Prueba de la sección Exportar Ficheros con extensión .pdf del caso de uso “Exportar Ficheros”.

<b>Sección Exportar Fichero con extensión .pdf.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>

EC 1.1 Exportar fichero con extensión .pdf correctamente.	El sistema permite exportar archivos .pdf.	Exporta un fichero .pdf con los datos de las columnas, profundidad, curvas, litologías y descripciones del proyecto activo.	El usuario escoge la opción exportar PDF. El sistema muestra la ventana de selección del directorio y del nombre del PDF. El usuario introduce los datos y presiona el botón Aceptar. El sistema crea el fichero .pdf con los datos de las columnas del proyecto activo.
EC 1.2 Exportar fichero con extensión .pdf sin proyectos creados.	No se exporta el fichero.	El sistema devolvió un error inesperado.	El usuario escoge la opción exportar PDF. El sistema muestra la ventana de selección del directorio y del nombre del PDF. El usuario introduce los datos y presiona el botón Aceptar. El sistema colapsó.

**Tabla 8** Caso de Prueba de la sección Exportar Ficheros con extensión .jpg del caso de uso “Exportar Ficheros”.

<b>Sección Exportar Fichero con extensión .jpg.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 2.1: Exportar fichero con extensión .jpg satisfactoriamente.	El sistema permite exportar archivos .jpg.	Exporta un fichero .jpg con los datos de las columnas, profundidad, curvas, litologías y descripciones del proyecto activo.	El usuario escoge la opción exportar imagen. El sistema muestra la ventana de selección del directorio y del nombre de la imagen. El usuario introduce los datos y presiona el botón Aceptar. El sistema crea el fichero .jpg con los datos de las columnas del proyecto activo.

EC 2.2 Exportar fichero con extensión .jpg sin proyectos creados.	No se exporta el fichero.	Muestra un mensaje avisando que no existen proyectos creados.	El usuario escoge la opción exportar imagen. El sistema muestra un mensaje avisando que no existen proyectos creados.
-------------------------------------------------------------------	---------------------------	---------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------

### 3.6.2.1.3 Caso de prueba para el caso de uso del sistema “Importar Ficheros”

Descripción General: El caso de uso se inicia cuando el usuario selecciona la opción importar datos de las curvas en el menú “Archivo” o en la barra de herramientas. Este caso de uso da paso a poder graficar las curvas contenidas en el fichero a importar.

Condiciones de Ejecución: Acceder al menú “Archivo” o a la barra de herramientas y escoger un fichero .las.

**Tabla 9** Caso de Prueba de la sección Importar Ficheros con extensión .las del caso de uso “Importar Ficheros”.

Sección Importar Fichero con extensión .las.			
Escenarios de la sección	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
EC 1.1 Importar fichero con extensión .las correctamente.	El sistema permite cargar los datos de las curvas contenidas en un fichero .las.	Muestra la ventana para seleccionar las curvas a importar.	El usuario escoge la opción importar datos de las curvas. El sistema muestra la ventana de selección del fichero. El usuario escoge el fichero y presiona el botón Aceptar. El sistema muestra la ventana para seleccionar las curvas a importar.
EC 1.2: Importar fichero con extensión .las sin proyectos creados.	No se importan ficheros.	Muestra un mensaje avisando que no existen proyectos creados.	El usuario escoge la opción importar datos de las curvas. El sistema verifica que hayan proyectos creados y muestra un mensaje avisando que no hay ningún proyecto creado.

### 3.6.2.1.4 Caso de prueba para el caso de uso del sistema “Construir Columna Curvas”

Descripción General: El caso de uso se inicia cuando el usuario luego de importar los datos de las curvas selecciona las curvas que desea graficar.

Condiciones de Ejecución: Haber importado un fichero .las.

**Tabla 10** Caso de Prueba de la sección Graficar Curvas del caso de uso “Construir Columna Curvas”.

<b>Sección Graficar Curvas.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1 Graficar curvas correctamente.	El sistema permite agregar nuevas curvas a la columna de las curvas del proyecto.	Muestra las gráficas correspondientes a las curvas seleccionadas.	El usuario escoge las curvas que desea graficar y presiona el botón Aceptar. El sistema muestra las gráficas correspondientes a las curvas seleccionadas sobre la columna de curvas del proyecto.
EC 1.2: Graficar curvas sobre una capa con curva asignada.	El sistema no grafica las curvas seleccionadas sobre una capa que ya contiene una curva asignada.	Muestra la curva seleccionada sobre la curva existente en esa capa, constituyendo esto una no conformidad.	El usuario escoge las curvas que desea graficar y presiona el botón Aceptar. El sistema grafica la curva seleccionada sobre la que estaba asignada a esa capa.

### 3.6.2.1.5 Caso de prueba para el caso de uso del sistema “Construir Columna Litológica”

Descripción General: El caso de uso se inicia cuando el usuario da clic o arrastra el mouse con el clic presionado sobre la columna litológica. Este caso de uso se realiza después de haber construido la columna de curvas.

Condiciones de Ejecución: Haber seleccionado una litología en el panel de símbolos.

**Tabla 11** Caso de Prueba de la sección Insertar Litologías del caso de uso “Construir Columna Litológica”.

<b>Sección Insertar Litologías.</b>
-------------------------------------

Escenarios de la sección	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
EC 1.1 Insertar litologías correctamente.	El sistema permite agregar nuevas imágenes de litologías a la columna litológica del proyecto.	Muestra la imagen seleccionada en la posición donde se dio clic con el mouse y muestra varias imágenes al arrastrar el mouse.	El usuario selecciona en el panel de símbolos una imagen correspondiente a una litología. El usuario da clic sobre la columna litológica. El sistema muestra la imagen seleccionada en la posición donde se dio clic. El usuario arrastra el mouse con el clic presionado sobre la columna litológica. El sistema muestra varias imágenes en las distintas posiciones del mouse.
EC 1.2: Insertar litologías sobre un área donde se encuentren otros símbolos.	El sistema permite agregar imágenes encima de otros símbolos presentes en la columna.	Muestra la imagen por encima de los símbolos que ya se encontraban en esa posición, constituyendo esto una no conformidad.	El usuario selecciona en el panel de símbolos una imagen correspondiente a una litología. El usuario da clic sobre la columna litológica donde existen símbolos. El sistema muestra la imagen por encima de los símbolos que ya se encontraban en esa posición.

**Tabla 12** Caso de Prueba de la sección Insertar Fósiles del caso de uso “Construir Columna Litológica”.

<b>Sección Insertar Fósiles.</b>			
Escenarios de la sección	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
EC 2.1: Insertar fósiles correctamente.	El sistema permite agregar nuevas imágenes de fósiles a la columna litológica del proyecto.	Muestra la imagen seleccionada en la posición donde se dio clic con el mouse.	El usuario selecciona en el panel de símbolos una imagen correspondiente a un fósil. El usuario da clic sobre la columna litológica. El sistema muestra el símbolo seleccionado en la posición donde se dio clic con el mouse.

EC 2.2: Insertar fósiles en un área donde se encuentra otro fósil.	El sistema no muestra imágenes de fósiles sobre otros que sean del mismo tipo.	No muestra la imagen seleccionada.	El usuario selecciona en el panel de símbolos una imagen de un fósil. El usuario da clic en la columna litológica. El sistema comprueba que el símbolo presente en la columna litológica es del mismo tipo que el que se desea insertar. El sistema no inserta el símbolo.
--------------------------------------------------------------------	--------------------------------------------------------------------------------	------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Tabla 13** Caso de Prueba de la sección Insertar Minerales del caso de uso “Construir Columna Litológica”.

<b>Sección Insertar Minerales.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1: Insertar minerales correctamente.	El sistema permite agregar nuevas imágenes de minerales a la columna litológica del proyecto.	Muestra la imagen del mineral en la posición donde se dio clic con el mouse.	El usuario selecciona en el panel de símbolos una imagen correspondiente a un mineral. El usuario da clic sobre la columna litológica. El sistema muestra el símbolo seleccionado en la posición donde se dio clic con el mouse.

**Tabla 14** Caso de Prueba de la sección Insertar Texturas del caso de uso “Construir Columna Litológica”.

<b>Sección Insertar Texturas.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1: Insertar texturas correctamente.	El sistema permite agregar nuevas imágenes de texturas a la columna litológica del proyecto.	Muestra la imagen de la textura en la posición donde se dio clic con el mouse.	El usuario selecciona en el panel de símbolos una imagen correspondiente a un mineral. El usuario da clic sobre la columna litológica. El sistema muestra el símbolo seleccionado en la posición donde se dio clic con el mouse.

**Tabla 15** Caso de Prueba de la sección Eliminar Símbolo del caso de uso “Construir Columna Litológica”.

<b>Sección Eliminar Símbolo.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1: Eliminar símbolo correctamente.	El sistema permite eliminar símbolos de la columna litológica del proyecto.	Elimina el símbolo seleccionado de la columna litológica.	El usuario selecciona el símbolo que desea eliminar dando clic sobre éste y presiona la tecla “delete”.  El sistema elimina el símbolo de la columna litológica.

### 3.6.2.1.6 Caso de prueba para el caso de uso del sistema “Construir Columna Descripciones”

**Descripción General:** El caso de uso se inicia cuando el usuario da doble clic sobre la columna descripciones, ya sea sobre un área vacía u ocupada por una descripción, este caso de uso permite realizar una breve descripción sobre las litologías presentes en la columna litológica. Mediante el movimiento del mouse sobre la columna el usuario puede apreciar la posición en donde puede insertar la descripción.

**Condiciones de Ejecución:** Dar doble clic sobre la columna de descripciones.

**Tabla 16** Caso de Prueba de la sección Nueva Descripción del caso de uso “Construir Columna Descripciones”.

<b>Sección Insertar Nueva Descripción.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1: Insertar nueva descripción correctamente.	El sistema permite adicionar una nueva descripción de una litología.	Muestra un campo de texto donde se pudo insertar una breve descripción.	El usuario presiona doble clic sobre la columna de descripciones el sistema.  El sistema muestra un campo de texto donde se inserta una breve descripción.  El usuario inserta el texto de la descripción y presiona clic fuera de ésta.  El sistema guarda la descripción.

**Tabla 17** Caso de Prueba de la sección Modificar Descripción del caso de uso “Construir Columna Descripciones”.

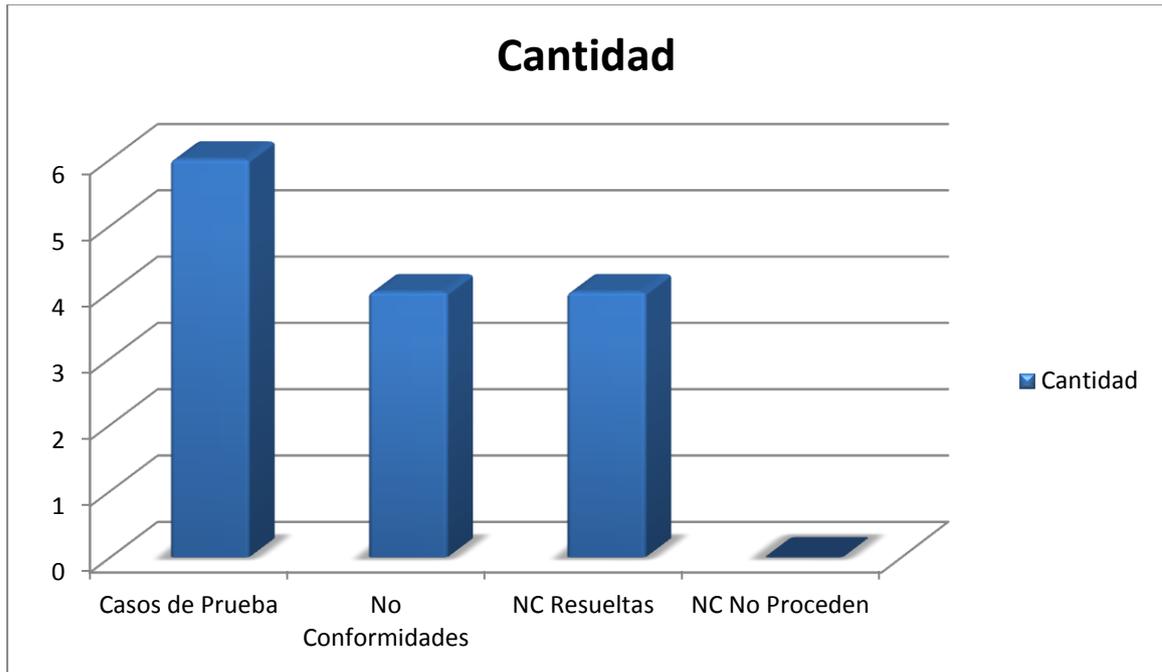
<b>Sección Modificar Descripción.</b>			
<b>Escenarios de la sección</b>	<b>Descripción de la funcionalidad</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>

<p>EC 1.1: Modificar descripción correctamente.</p>	<p>El sistema permite modificar una descripción de la columna de descripciones.</p>	<p>Muestra un campo de texto en la posición con “y” igual a la de la descripción y con el texto de la misma, donde se pudo realizar cambios.</p>	<p>El usuario presiona doble clic sobre una descripción de la columna de descripciones el sistema.</p> <p>El sistema muestra un campo de texto en la posición con “y” igual a la de la descripción y con el texto de la misma.</p> <p>El usuario realiza los cambios necesarios y presiona clic fuera de la descripción.</p> <p>El sistema guarda los cambios efectuados.</p>
-----------------------------------------------------	-------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Tabla 18** Caso de Prueba de la sección Eliminar Descripción del caso de uso “Construir Columna Descripciones”.

Sección Eliminar Descripción.			
Escenarios de la sección	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
<p>EC 1.1: Eliminar descripción correctamente.</p>	<p>El sistema permite eliminar una descripción presente en la columna de descripciones.</p>	<p>Elimina la descripción seleccionada de la columna de descripciones.</p>	<p>El usuario presiona clic sobre la descripción que desea eliminar y presiona la tecla “delete”.</p> <p>El sistema elimina la descripción de la columna de descripciones.</p>

Se realizaron las pruebas de caja negra que brindarán al cliente conformidad y seguridad ante las funcionalidades del sistema. Se realizó un caso de prueba por cada caso de uso arrojando los resultados que se presentan en el siguiente gráfico:



**Figura 25** Pruebas de Caja Negra

### 3.7 Conclusiones Parciales

Redefinir el modelo de diseño del sistema en la etapa de implementación y definir un estándar de codificación brindó claridad a la hora de programar y permitió el desarrollo del software en el tiempo esperado.

Las pruebas de caja negra sirvieron para corregir errores presentes en el software, logrando poner éste en condiciones de ser desplegado en las entidades petroleras de nuestro país, lo que permitirá facilitar el trabajo a la hora de construir la columna litológica de los pozos de petróleo en perforación.

## **CONCLUSIONES GENERALES**

Con la realización de este trabajo se pueden afirmar las siguientes conclusiones:

- ✓ Se logró desarrollar un Sistema para Graficar las Columnas Litológicas de los Pozos de Petróleo, siendo éste completamente libre para la empresa CUPET, lo cual da cumplimiento a los objetivos planteados en este trabajo.
- ✓ El uso de la biblioteca Qt y del estilo de programación Arquitectura en Capas, permitió elaborar subsistemas que pueden convertirse en componentes reutilizables para aplicaciones de similares características.
- ✓ El presente trabajo aporta una solución factible a la situación problemática que lo originó y su explotación significará una mejora considerable en la producción de la empresa CUPET de nuestro país.
- ✓ La validación del sistema mediante las pruebas de caja negra demostró la robustez de la solución y permitió corregir algunas deficiencias presentes en el software, perfeccionando así el funcionamiento del sistema.

## RECOMENDACIONES

Con el objetivo de incorporar mejoras significativas al Sistema para Graficar Columnas Litológicas de Pozos de Petróleo se recomienda:

- ✓ Incrementar los tipos de extensiones que pueden tener los ficheros con los que trabaja el sistema.
- ✓ Añadir las funcionalidades Gestionar usuarios y Autenticar usuarios con el objetivo de garantizar una mejor seguridad del sistema.
- ✓ Mejorar la precisión del escalado de imágenes a la hora del cambio de escala de un proyecto creado en el sistema, para lograr resultados con una mayor aproximación a la realidad.
- ✓ Probar el sistema en varias instalaciones petroleras que requieran del mismo y en presencia de especialistas en sedimentología con el objetivo de comprobar que el software se encuentra en condiciones de ser utilizado por la empresa CUPET.

## REFERENCIAS BIBLIOGRÁFICAS

1. Perspectiva Mundial. [En línea] Julio de 2004. [Citado el: 2011 de Febrero de 12.] <http://www.perspectivamundial.com/2004/2807/280715.shtml>.
2. **Cruz, Eduardo Ramírez.** *Análisis y Diseño del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, 2010. Tesis de Grado.
3. **Cuevas, David Tavares.** *ANÁLISIS Y DISEÑO DEL SISTEMA DE MANEJO INTEGRAL DE PERFORACIÓN DE POZOS.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, 2009. Tesis de Grado.
4. **Allen Paneque, Jorge Roberto.** *Módulo Pozo del Sistema de Manejo Integral de Perforación de Pozos.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, 2010. Tesis de Grado.
5. **Rodríguez, Milton Benito León.** *Diseño del Segundo Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, 2010. Tesis de Grado.
6. Diccionario de la LENGUA ESPAÑOLA - Vigésima segunda edición. [En línea] 13 de Diciembre de 2006. [Citado el: 16 de Febrero de 2011.] <http://buscon.rae.es/drae/>.
7. **Hernández León, Rolando Alfredo y Coello González, Sayda.** *El Paradigma Cuantitativo de la Investigación Científica.* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n, 2002. Tesis de Grado. EDUNIV, 2002. 959-16-0343-6.
8. **Álvarez González, Maikel Hugo.** *Diseño de la base de datos del Sistema de Información de Perforación de Pozos (SIPP).* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n, 2009. Tesis de Grado.
9. QED. Environmental Systems. [En línea] [Citado el: 10 de Febrero de 2011.] <http://cee.com/products/sampling/WellWizard/WellWizard.html>.
10. Wellsight Systems. [En línea] 1994-2008. [Citado el: 2011 de Febrero de 20.] <http://www.wellsight.com>.
11. Modelaje de Pozos. *Modelaje de Pozos.* [En línea] [Citado el: 20 de Febrero de 2011.] <http://modelaje-de-pozos.blogspot.com/search/label/WellFlo>.
12. AddLink. *AddLink.* [En línea] 01 de Noviembre de 2009. [Citado el: 22 de Febrero de 2011.] <http://www.addlink.es/productos.asp?pid=430>.

13. Symfony. *HowToContributeToSymfony*. [En línea] [Citado el: 25 de Febrero de 2011.] <http://trac.symfony-project.org/wiki/HowToContributeToSymfony#CodingStandards>.
14. **Infosintesis Solutions Group, S.L.** Formación, asesoramiento y desarrollo de proyectos informáticos . [En línea] [Citado el: 26 de Febrero de 2011.] [www.infosintesis.net..](http://www.infosintesis.net..)
15. **Suarez, Jose Gregorio.** [En línea] 28 de Enero de 2009. [Citado el: 26 de Febrero de 2011.] [www.Los-patrones-de-diseno.htm](http://www.Los-patrones-de-diseno.htm).
16. **Frank Buschmann, y otros, y otros.** *Oriented Software Architecture. A System of Patterns*. 1996.
17. **Llorente, César de la Torre , y otros, y otros.** *Arquitectura N-Capas Orientada al Dominio - Microsoft Architecture*. s.l. : Primera Edición, Noviembre - 2010.
18. **Morales, Frank Hector Rios.** *Arquitectura de Software del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, 2010. Tesis de Grado.
19. **Uwe Rathmann y Josef Wilgen.** *Qwt - Qt Widgets for Technical Applications*. [En línea] [Citado el: 15 de Abril de 2011.] <http://qwt.sourceforge.net/>.
20. **Giron, Marco Antonio Pereira.** [www.scribd.com](http://www.scribd.com). [En línea] 2007. [Citado el: 01 de Marzo de 2011.] <http://www.scribd.com/doc/2080534/UML>.
21. Metodología de Desarrollo de Software. [En línea] <http://www.scribd.com/doc/12983329/Metodologia-de-Desarrollo-de-Software>.
22. [mitecnologico.com](http://www.mitecnologico.com). [En línea] [Citado el: 25 de Marzo de 2011.] <http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos>.
23. Enterprise Architect. [En línea] [Citado el: 02 de Abril de 2011.] [http://www.sparxsystems.com.ar/resources/tutorial/use\\_case\\_model.html](http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html).
24. **González, Rolando Ávila.** *Componente para visualizar curvas de registro de pozos*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, Febrero de 2010. Tesis de Grado.
25. **Larman, Craig.** *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Mexico . s.l. : s.n, 1999. 970-17-0261-1.

26. **Rodríguez, Yusleidy Neira.** *Análisis y Diseño del Sistema para el análisis petrofísico.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n, 2009. Tesis de Grado.
27. **S. Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* Madrid : Quinta Edición, 2001.

## **GLOSARIO DE TÉRMINOS**

**API:** Una interfaz de programación de aplicaciones o API (del inglés application programming interface) es el conjunto de funciones y procedimientos (métodos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**CEINPET:** Centro de Investigaciones de Petróleo.

**CUPET:** Unión Cuba-Petróleo.

**DIPP:** Dirección de Intervención y Perforación de Pozos.

**Geología:** Ciencia que estudia y describe los materiales que forman el globo terrestre, las transformaciones y evolución de la Tierra y los fósiles.

**GPL:** Es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

**Las:** (Log ASCII Standard). Es un formato de archivo estándar común en el petróleo y el gas de la industria para almacenar la información de los registros de pozo.

**MINBAS:** Ministerio de la Industria Básica.

**POO:** Programación Orientada a Objetos.

**Qt:** Es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores.

**Sedimentología:** Estudio y análisis de los sedimentos (materia que, habiendo estado suspensa en un líquido, se posa en el fondo por su mayor gravedad).

**SIPP:** Sistema de Manejo Integral de Perforación de Pozos.

**TIC:** Tecnología de la Información y las Comunicaciones.

**UCI:** Universidad de las Ciencias Informáticas.