

**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN**

**Reingeniería e implementación
de nuevos componentes para la
aplicación de lectura y visualización
de registros acústicos y resistivos
(VenLog 2.0)**



Trabajo Especial de Grado presentado ante
la ilustre Universidad Central de Venezuela por:
Br. Darwin J. Martínez L.

Para optar por el título de Licenciado en Computación

Tutora:
Dra. Alecia Eleonora Acosta

Octubre, 2012

Resumen

El mantener un hoyo estable es una de las principales tareas de perforación de la industria petrolera, ya que los problemas de inestabilidad asociados al pozo resultarán altamente costosos y tendrán un impacto severo en el cronograma de perforación. La evaluación de estabilidad se realiza a partir de la interpretación de perfiles de pozos, la cual es una técnica utilizada en la industria petrolera para grabar propiedades roca-fluidos y encontrar zonas de hidrocarburos en las formaciones geológicas dentro de la corteza terrestre. Estos perfiles son almacenados en archivos de tipo DLIS (*Digital Log Interchange Standard*), los cuales cumplen un estándar sintáctico para sísmica, perforación y perfilaje de pozos, estos archivos no son legibles en editores de texto convencionales por lo que se necesita de un software especializado para su interpretación. Lo anterior motivó el desarrollo de una herramienta de software que sea capaz de procesar perfiles sísmicos, resistivos, y de imagen, almacenados en archivos de tipo DLIS, con el propósito de apoyar el estudio de estabilidad de hoyos realizado por parte de la Gerencia Técnica de Producción y Construcción de Pozos de PDVSA Intevep. Para el desarrollo de esta herramienta fue necesario el estudio a fondo del estándar DLIS usando MeRinde como modelo de proceso para lograr obtener un software de calidad, y se obtuvo como resultado VenLog 2.0. Esta herramienta está fundamentada en estándares corporativos, orientada a apoyar el proceso de construcción de pozos, específicamente en las tareas de estabilidad de hoyos.

Palabras Claves

Perfilaje, Perfiles sísmicos, perfiles resistivos, perfiles de imagen, DLIS, Estabilidad de hoyos.

Dedicatoria

**A mis padres, por guiarme en la crianza para
convertirme en la persona que soy ahora.**

Agradecimientos

A Dios, por haberme dado la sabiduría y la fortaleza para alcanzar esta meta.

A mi Familia, por apoyarme incondicionalmente y ser un ejemplo constante de trabajo y perseverancia.

Al grupo docente de la escuela de Computación, por contribuir en mi formación profesional.

A la Prof. Eleonora Acosta, quien con su orientación, enseñanza y apoyo hizo posible la culminación de este proyecto.

A la Gerencia de Automatización, Informática y Telecomunicaciones de PDVSA-Intevep, por asignarme este trabajo y confiar en mi capacidad para realizarlo. En especial agradezco a la Ing. Carmen Salazar, por su asesoría y excelente guía desde el inicio de este trabajo.

A Gabriela, por ser la principal responsable de que este trabajo llegara a buen término, estando incondicionalmente a mi lado, animándome siempre a continuar.

A todas aquellas personas que no nombré, pero que forman parte de mi vida y me han apoyado en todo momento, de todo corazón gracias.

Índice de contenido

Introducción.....	5
Capítulo 1Fundamentos teóricos del perfilaje de pozos	
	9
1.1 Perfilaje de pozos	9
1.1.1 Perfil eléctrico	11
1.1.2 Perfil Sónico.....	11
1.1.3 Perfil de imagen	12
1.2 Estándar DLIS.....	17
1.2.1 Organización de los datos en un archivo DLIS.....	18
Capítulo 2Desarrollo de la aplicación VenLog 2.0 siguiendo el modelo de procesos MeRinde.....	24
2.1 Modelo de proceso MeRinde	24
2.2 Estructura de MeRinde.....	25
2.2.1 Requerimientos	26
2.2.1.1	Requerimientos no funcionales
	26
2.2.1.2	Requerimientos funcionales
	27
2.2.2 Análisis y diseño	31
2.2.3 Implementación.....	32
2.2.4 Pruebas.....	38
Capítulo 3Escenarios de uso de la herramienta VenLog 2.0	
	46
3.1 Instalación y desinstalación de VenLog 2.0.....	46
3.2 Escenario de uso 1: Graficar curva(s)	55
3.3 Escenario de uso 2: Graficar imagen	58
3.4 Escenario de uso 3: Graficar imagen resistiva	60
Conclusiones y recomendaciones.....	63
Referencias bibliográficas	65
Anexo 1.....	67
Anexo 2.....	92

Anexo 3.....	114
Anexo 4.....	118
Anexo 5.....	124

Índice de figuras

Figura 1-1. Representación de la pared del hoyo (Díaz, 2006).....	12
Figura 1-2. Toma de datos de con una herramienta sónica (Baker, 2006).	13
Figura 1-3. Perfil tomado con la herramienta CBIL de la compañía de servicios Baker Hughes (Baker, 2008).....	14
Figura 1-4. Toma de datos de con una herramienta resistiva (Díaz, 2006).....	16
Figura 1-5. Registro de imagen con una herramienta resistiva (Díaz, 2006).	17
Figura 1-6. Organización de los datos en un DLIS (Chumbimuni, 2008).	18
Figura 1-7. Estructura de un EFLR (POSC, 1998).....	20
Figura 1-8. Atributos del objeto origen (POSC, 1998).....	21
Figura 1-9. Atributos del objeto canal (POSC, 1998).	22
Figura 1-10. Atributos del objeto trama (POSC, 1998).....	22
Figura 2-1. Esfuerzo en actividades según las fases de proyecto de MeRinde (Marrero & Santos, 2007).	25
Figura 2-2. Diagrama de casos de uso de VenLog 2.0 (módulo lector de registros).	28
Figura 2-3. Diagrama de componentes VenLog 2.0.....	31
Figura 2-4. Diagrama de clase capa presentación VenLog 2.0.	32
Figura 2-5. Estructura de la plataforma de desarrollo.	33
Figura 2-6. Integración del modelo de proceso con las herramientas de software en la disciplina implementación.....	37
Figura 2-7. Integración de la metodología con las herramientas de software en la disciplina pruebas.....	44
Figura 3-1. Ventana de bienvenida del asistente de instalación de VenLog 2.0.	47
Figura 3-2. Ruta de instalación.....	47
Figura 3-3. Extracción de archivos.....	48
Figura 3-4. Finalizando la instalación.	48
Figura 3-5. Asistente de desinstalación.	49
Figura 3- 6. Desinstalación de VenLog 2.0.	50

Figura 3-7. Finalizando la desinstalación.	50
Figura 3- 8. Marco de trabajo de VenLog 2.0.	51
Figura 3-9. Ventana emergente que permite indicar la ruta del archivo a procesar.	53
Figura 3- 10. Panel Trama.	53
Figura 3-11. Panel Origen.	54
Figura 3-12. Entorno de VenLog 2.0 con un archivo abierto.	54
Figura 3-13. Selección de canales (Curva).	55
Figura 3-14. Menú de las curvas.	56
Figura 3- 15. Cambiar mostrar cuadrícula y cambiar color de la curva.	56
Figura 3-16. Cambiar escala vertical.	57
Figura 3- 17. Curvas superpuestas.	58
Figura 3-18. Selección de canales (Imagen).	59
Figura 3- 19. Cambiar paleta de colores de la imagen.	59
Figura 3-20. Selección de canales (Imagen resistiva).	60
Figura 3-21. Selección de canales (curvas e imágenes).	61
Figura 3- 22. Selección de canales (curvas, imágenes e imagen resistiva).	62

Índice de tablas

Tabla 1-1. Perfiles más usados por la Industria Petrolera (Smolen, 1995).	10
Tabla 1-2. Distribución de la SUL (POSC, 1998).....	19
Tabla 2-1. Descripción caso de uso Leer archivo.....	29
Tabla 2-2. Descripción caso de uso Generar curva.	30
Tabla 2-3. Descripción caso de uso Generar imagen.	30
Tabla 2-4. Herramientas de software para la ejecución de las pruebas.....	38
Tabla 2-5. Herramientas de hardware para la ejecución de las pruebas.....	38
Tabla 2-6. Configuración del ambiente de pruebas.....	39
Tabla 2-7. Pruebas de integración.	39
Tabla 2-8. Resultados de pruebas de integración.	40
Tabla 2-9. Pruebas de sistema.	41
Tabla 2-10. Resultados de pruebas de sistema.	42
Tabla 2-11. Pruebas de aceptación.	43
Tabla 2-12. Resultados de pruebas de aceptación.....	43

Introducción

PDVSA Intevep es el brazo tecnológico de Petróleos de Venezuela, una filial de investigación cuya orientación estratégica es generar soluciones tecnológicas integrales, con especial énfasis en las actividades de Exploración, Producción, Refinación e Industrialización. De igual manera, es el ente responsable del resguardo del acervo tecnológico de PDVSA. Así mismo, desarrolla tecnologías propias en áreas con oportunidades diferenciales, impulsa la cooperación e integración con el sector técnico-científico e industrial de Venezuela y asegura, al mismo tiempo, la correcta gestión ambiental en las operaciones de PDVSA (PDVSA, 2005).

La Gerencia Técnica de Producción y Construcción de Pozos (PRCP) de PDVSA Intevep, se encarga de generar los productos tecnológicos requeridos por PDVSA, asegurando eficiencia y efectividad en la ejecución de proyectos de investigación, desarrollo y asistencia técnica. En el marco de esta gerencia y con el apoyo de la Fabrica de Software de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT) localidad Miranda Oeste de PDVSA, se está desarrollando el proyecto *Suite de Perforación*, cuyo objetivo es desarrollar tecnologías propias, y reemplazar progresivamente las tecnologías foráneas en aplicaciones de software para la construcción y mantenimiento de pozos, en el contexto de los lineamientos gubernamentales y corporativos de desarrollo endógeno y soberanía tecnológica (Zambrano, 2011).

Una de las tareas de la gerencia de PRCP es el estudio de estabilidad de hoyos, esta tarea es una de las más importantes durante la perforación de un pozo, ya que los problemas de inestabilidad asociados a este resultarán altamente costosos y tendrán un impacto severo en el cronograma de perforación. La evaluación se realiza a partir de la interpretación de perfiles de pozos, que no es más que una técnica utilizada en la industria petrolera para grabar propiedades roca-fluidos y encontrar zonas de hidrocarburos en las formaciones geológicas dentro de la corteza terrestre. Este proceso consiste en colocar una herramienta de perfilaje al extremo final de un cable e introducirla dentro de un pozo para medir las propiedades de las rocas y los fluidos de las formaciones, generando de esta manera un registro que debe ser interpretado por un software especializado.

Las compañías de servicio petroleras como: Baker Hughes, Halliburton, Schlumberger y Waterford Oil Co, entre otras, son contratadas por PDVSA para realizar trabajos de perforación y mantenimiento de pozos, y como parte de sus actividades esta la realización del perfilaje e interpretación de los mismos. Sin embargo las interpretaciones de estos, en algunas ocasiones, presentan discrepancia entre los datos mostrados en el informe entregado a los analistas de PDVSA y el estado actual del pozo. Es por esto que los

científicos e ingenieros especializados en estabilidad de hoyo, desean realizar sus propios análisis e interpretaciones a partir de herramienta de software que les permita visualizar el estado del hoyo, a través de las imágenes de registros sísmicos y resistivos, complementando el análisis con los cálculos adecuados que permiten determinar rupturas y fallas en el hoyo.

De esta manera surge la iniciativa VenLog, como una herramienta de software basada en las reglas de negocio suministrada por los analistas de PDVSA Intevep y desarrollada por un ente externo contratado por PDVSA. Esta herramienta debía realizar las siguientes tareas: lector de archivos LAS v3.0, lector DLIS v1, modelado de estructuras geológicas, inteligencia en deformaciones de pozo, procesamiento de registros, procesamiento de registros de imágenes e inteligencia en cementación.

VenLog fue desarrollado con el lenguaje de programación C#, sobre plataforma .NET, utilizando la biblioteca VTK (*Visualization Toolkit*) especializada en la visualización y procesamiento de gráficos.

Esta primera versión de VenLog, no pudo ser puesto en producción¹, ya que no pasó la fase de pruebas realizada por la Gerencia de AIT localidad Miranda Oeste. Por lo tanto, y para no perder el trabajo previo, se realizaron evaluaciones exhaustivas para identificar los principales problemas de la herramienta y determinar la posibilidad de solventar las fallas encontradas o desechar definitivamente la herramienta de software. De esta manera se tomó como antecedente a este Trabajo Especial de Grado un Trabajo de Pasantía, el cual consistió en realizarle a VenLog la evaluación mencionada anteriormente y cuyos resultados arrojaron el planteamiento de una reingeniería, basado en un nuevo levantamiento de requisitos y reutilización de algunos sus componentes. Los componentes que son reutilizables de VenLog son: libNumber, que es una librería de Software Libre que se encuentra disponible en la web, desarrollada por John Jiyang Hou en el año 2005, la cual se encarga de transformar una expresión de bytes a un número de punto flotante o viceversa; y las librerías `vvm_data_log` y `vvm_data_node`, que fueron creadas por el grupo de desarrolladores del VenLog inicial, las cuales forman una estructura base eficiente para almacenar los datos obtenidos del archivos de entrada (Martínez 2012).

Después de analizar los resultados de la evaluación de VenLog, la gerencia de PRCP decide desarrollar una nueva herramienta de software, denominada VenLog 2.0 en dos (2) fases, la primera consiste en la implementación de un intérprete de archivos DLIS y la construcción de gráficas asociadas al archivo, y la segunda abarca el desarrollo de un conjunto de cálculos y operaciones sobre las gráficas construidas en la primera fase. Partiendo de esto

¹ Puesta en Producción: en PDVSA, la Gerencia de AIT define como puesta en producción la acción de validar la calidad de un software, y luego de aprobada esta etapa, colocarla en repositorios comunes que permiten su distribución en todas las localidades donde se requiera su instalación, es decir, se masifica el uso del software dentro de la empresa.

se consideró factible el desarrollo de la primera fase del proyecto para ser introducida como Trabajo Especial de Grado.

El objetivo general de este Trabajo Especial de Grado es desarrollar el módulo Lector de Registros DLIS capaz de procesar perfiles sísmicos, resistivos, y de imagen, para la herramienta de software VenLog en su versión 2.0, dando apoyo al estudio de Estabilidad de Hoyos.

VenLog 2.0 ajusta y unifica la forma del procesamiento de los datos dentro del programa, independientemente de la fuente propietaria: Baker, Schlumberger, Halliburton, o cualquier otra compañía de servicio. De igual modo, a pesar de que este trabajo está enfocado en la interpretación de registros sísmicos, resistivos y de imagen, esta aplicación podrá interpretar cualquier registro de tipo DLIS, apoyando de esta manera la optimización del análisis e interpretación de estos registros ya que elimina la dependencia de terceros y amplía e impulsa la ejecución de nuevos proyectos tanto en el área de investigación y desarrollo como asistencia técnica especializada. Cabe destacar, que la aplicación sólo permite generar las gráficas correspondientes a los datos seleccionados por el usuario, cambiar el color de la gráfica generada, aumentar o disminuir su tamaño para un mayor nivel de detalle y superponer las gráficas, sólo en el caso de las curvas, para hacer comparaciones entre estas. Es justo decir que, en el alcance del proyecto no está previsto que la herramienta realice cálculos asociados a fallas y fracturas que estén presente en la estructura del pozo, ya que como se mencionó anteriormente estos cálculos están previstos para el desarrollo de la segunda fase del proyecto.

Para el desarrollo de este Trabajo Especial de Grado, primeramente se hizo una revisión de los resultados obtenidos en el trabajo de pasantías, haciendo énfasis en la investigación del estándar DLIS (*Digital Log Interchange Standard*), el cual es un formato de archivo para el almacenamiento de perfiles de pozos de hidrocarburos.

Posteriormente se realizó un repaso de la revisión de los conceptos básicos sobre el estudio de estabilidad de hoyos, tales como, perfilaje de pozo, técnicas de medición, representación gráfica, tipos de perfiles, importancia, utilidad de estos en el estudio de estabilidad de hoyos, combinando esto con entrevistas a los especialistas en el área, para recabar el conocimiento necesario para el entendimiento del dominio del problema.

Se desarrollaron los componentes lógicos y de presentación para la primera fase de VenLog 2.0. Esta herramienta de software fue desarrollada bajo el paradigma de programación orientado a objetos y teniendo como guía el modelo de procesos MeRinde, permitiendo este último mantener la integridad del sistema mediante la coordinación de las actividades del ciclo de vida del software y quedando documentado el proceso en los diferentes artefactos

propuestos por disciplinas. Estos últimos se pueden ver con mayor detalle en los Anexos 1, 2 y 5.

La herramienta se implementó utilizando los estándares de arquitectura, lenguaje de programación plataforma y modelo de desarrollo, que se están utilizando actualmente en el desarrollo del proyecto *Suite de Perforación*, debido a que la herramienta formará parte del conjunto de aplicaciones que la conforman.

El presente Trabajo Especial de Grado está conformado además de la Introducción, las Conclusiones, Referencias bibliográficas y Anexos; por tres (3) capítulos de contenido descritos a continuación: El Capítulo 1 describe los fundamentos teóricos necesarios para entender el dominio del problema. El Capítulo 2 describe el desarrollo de la herramienta siguiendo el modelo de procesos MeRinde. El Capítulo 3 muestra varios recorridos por VenLog 2.0 mediante la descripción de tres (3) escenarios de uso.

Capítulo 1

Fundamentos teóricos del perfilaje de pozos

En este capítulo, se describen algunos de los fundamentos teóricos necesarios para entender la situación problemática a resolver con el desarrollo de este Trabajo Especial de Grado. Se define perfilaje, que es la técnica para generar el registro o perfil de pozo. Se enumeran los perfiles más utilizados y se describe la utilidad de algunos de estos. Por último se define el estándar DLIS, el cual es un formato de archivo utilizado para el almacenamiento de registros o perfiles de pozo. Los conceptos anteriores serán estudiados para ofrecer al lector una visión general sobre el dominio del problema.

1.1 Perfilaje de pozos

El perfilaje de pozos es una actividad muy importante dentro de la exploración y producción de hidrocarburos (petróleo y gas), la cual consiste en la toma y monitoreo de los perfiles o registros del pozo. Un registro o perfil de pozo es una grabación contra profundidad de alguna de las características de las formaciones rocosas atravesadas, hechas por aparatos o herramientas de medición en el hoyo del pozo (Schlumberger, 1966).

La principal función del perfilaje de pozo es la localización y evaluación de los yacimientos de hidrocarburos, ya que además, de brindar información sobre los fluidos presentes en el pozo (agua, petróleo o gas), permite conocer el estado actual de las paredes de este y determinar las posibles fallas o fracturas presentes en la estructura.

De acuerdo a las mediciones realizadas en el pozo, los perfiles pueden ser tomados en hoyo desnudo o entubado, dependiendo del momento de correr este perfil y los objetivos buscados (Smolen, 1995).

A continuación se describirán los distintos hoyos donde pueden ser tomados los perfiles:

Perfiles en hoyo desnudo

El objetivo de los perfiles en hoyo desnudo es medir las propiedades físicas, eléctricas, acústicas y radioactivas de las rocas; para luego, realizar una interpretación y así determinar las propiedades de la formación, tales como porosidad, permeabilidad, saturaciones, arcillosidad, entre otras. Estas propiedades son indispensables para la descripción y la

racional explotación del yacimiento. Los dispositivos de los perfiles que se corren en hoyo desnudo son diseñados específicamente para esa condición, la mayoría de estos no pueden ser corridos en hoyos entubados (Smolen, 1995).

En la Tabla 1-1 se presenta un resumen de los perfiles más usados en la industria petrolera.

Tipos de Perfil	Perfil
Perfil de diámetro de Pozo	-Bit Size (BS) -Caliper (CALI)
Perfil Eléctrico	-Potencial Espontáneo (SP) -Resistividad (Resistivity)
Perfil Radioactivo	-Rayos Gamma (RG) -Rayos Gamma Espectral (NGS)
Perfil de Porosidad	-Neutrón (CNL) -Densidad (FDC) -Sónico (BHC)
Perfil de Buzamiento	-Dip Meter (GR)
Perfil de resonancia Magnética	-Perfil de resonancia magnética nuclear (CRM)
Perfil de Imagen	-Perfil de imagen sónico -Perfil de imagen resistivo

Tabla 1-1. Perfiles más usados por la Industria Petrolera (Smolen, 1995).

Perfiles en hoyo entubado

Los perfiles en hoyo entubado son obtenidos mediante dispositivos diseñados específicamente para este fin. En general existen tres tipos de perfiles que se corren en pozos entubados:

- Registros de Cementación que determinan la calidad de la cementación del revestidor.
- Registros de Producción que miden la tasa y la naturaleza de los fluidos de los pozos en producción.
- Registros Equivalentes a los Corridos en Hoyo Desnudo, que tienen como fin obtener información adicional de las propiedades de la formación (Smolen, 1995).

Debido a que este Trabajo Especial de Grado está desarrollado para la Gerencia Técnica de Producción y Construcción de Pozos (PRCP) de PDVSA Intevep, específicamente para el grupo de Estabilidad de Hoyos y estos, para sus estudios sólo utilizan los perfiles en hoyo desnudo esta investigación se enfocará en este tipo de perfil, específicamente en los perfiles eléctricos de tipo resistivo, perfiles sónicos y perfiles de imagen, los cuales se definirán a continuación.

1.1.1 Perfil eléctrico

La exploración de superficie por medio de métodos eléctricos se viene realizando desde principios del siglo pasado para localizar depósitos minerales, efectuar mapeos geológicos y otras aplicaciones. En la segunda mitad de la década de los años 20, estos métodos de exploración fueron adaptados al perfilaje de las formaciones atravesadas por pozos. El registro de la resistividad en función de la profundidad, conocido como perfilaje eléctrico, fue pronto considerado un relevante método geológico y, a veces, un método de búsqueda de petróleo (Guyod, 1974).

A continuación se definirá el concepto de resistividad y su aplicación en los perfiles eléctricos.

Resistividad

La resistividad de un material es aquella propiedad que se opone a la circulación de la corriente eléctrica. Las medidas de resistividad registradas en un pozo perforado pueden utilizarse para determinar contactos de formaciones y obtener información sobre la naturaleza de las capas atravesadas por la perforación.

El perfil de resistividad puede obtenerse introduciendo y haciendo descender uno o varios electrodos dentro del pozo, haciendo las medidas por medio de arreglos eléctricos con el equipo apropiado. Estas medidas son registradas en función de la profundidad, denominándose a la gráfica resultante “Curva de Resistividad”. Los valores de resistividad están expresados en ohm-m².

Existen dos tipos principales de perfiles resistivos: el perfil lateral (LateroLog), que se utiliza en lodos conductivos (lodo salado), y el perfil de inducción (InductionLog) que se utiliza en lodos resistivos (lodo fresco o base aceite) (Schlumberger, 1970).

1.1.2 Perfil Sónico

El perfil sónico es la medición de la velocidad del sonido dentro del pozo, fue originalmente concebido como una ayuda para la interpretación sísmica. Desde entonces ha tenido más aceptación como herramienta básica para la determinación de porosidad, fracturas, determinación de litología y correlaciones básicas.

Actualmente las herramientas para el perfilaje sónico permiten la digitalización del tren de ondas completo en el fondo, de tal manera que se elimina la distorsión del cable. De este

² ohm-m: unidad derivada de resistencia eléctrica por metro.

modo, la mayor capacidad de obtención y procesamiento de datos permite el análisis de todos los componentes de la onda de sonido (Tittman. 1986).

Los perfiles sísmicos son de gran ayuda ya que a través de estos es posible realizar diferentes estudios, entre los cuales se encuentran: correlación de datos sísmicos, sismogramas sintéticos, determinación de porosidad, detección de gas, detección de fracturas, características mecánicas de la roca y estabilidad de hoyo.

1.1.3 Perfil de imagen

Los perfiles de imagen proveen información de las paredes del pozo y permiten la descripción del yacimiento mediante la identificación de rasgos estratigráficos, zonas fracturadas, orientación de fracturas, cambios en la densidad de la roca, estratos delgados, análisis estructural y orientación de los esfuerzos horizontales (Díaz, 2006).

La presentación estándar para el registro de imagen del hoyo, es el formato de manera desplegada *unwrapped borehole*. La imagen cilíndrica de la superficie del hoyo es desplegada por el azimut³ norte y desenrollada para tener un listón plano como se muestra en la Figura 1-1.

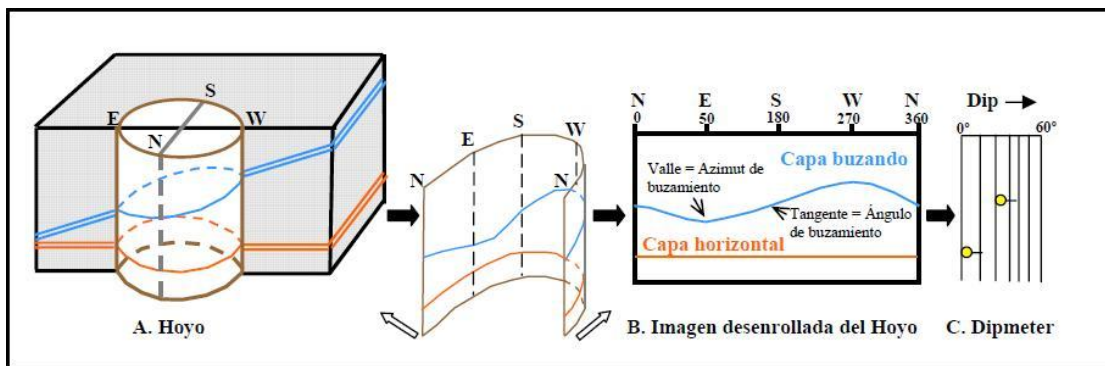


Figura 1-1. Representación de la pared del hoyo (Díaz, 2006).

Los perfiles de imagen más utilizados en la actualidad se clasifican de acuerdo con su principio de operación en dos grupos: sísmico y resistivo; los cuales se definen a continuación:

³ Azimut: Es el ángulo de una dirección, contado en el sentido de las agujas del reloj a partir del norte geográfico.

- **Perfil sónico de imagen**

Los perfiles sónicos de imagen proporcionan información de la pared del pozo y permiten la descripción de varias propiedades del yacimiento por medio de la identificación de características estratigráficas, intervalos y orientación de las fracturas, cambios en la porosidad de la roca, litofacies, estratos delgados, análisis estructural, y orientación de esfuerzos horizontales (Baker, 2006). La Figura 1-2 ilustra cómo la herramienta sónica realiza el proceso de perfilaje.

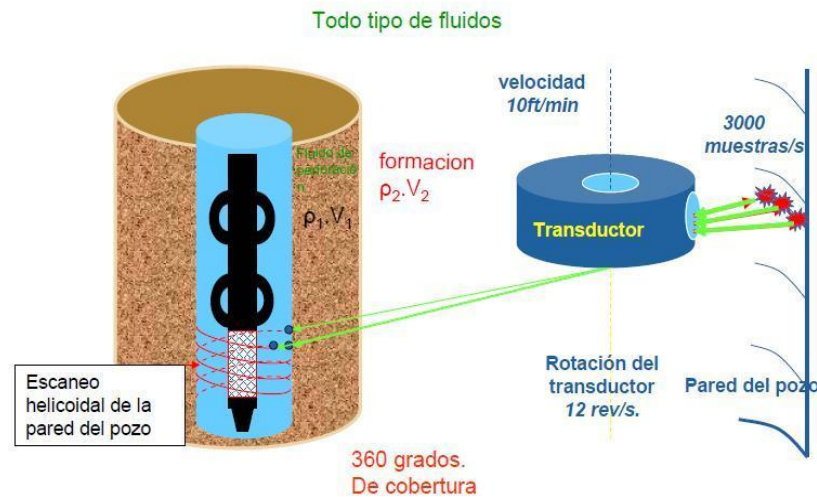


Figura 1-2. Toma de datos de con una herramienta sónica (Baker, 2006).

En la Figura 1-2 se puede observar la manera en que la herramienta se va desplazando a través del hoyo, girando sobre su propio eje, mientras envía señales sónicas, las cuales al volver, indican el tiempo de tránsito y la amplitud.

Los perfiles de imagen son utilizados para determinar la caracterización de los sistemas de fractura, orientación de los esfuerzos horizontales y facies sedimentarias, a continuación se explican cada uno de estos.

- o **Caracterización de los sistemas de fractura**

Los perfiles sónicos permiten definir aquellas características del sistema de fracturas que influyen en la productividad del pozo. Se pueden contar como tales características: la orientación, densidad, apertura y distribución de las fracturas.

La orientación de las fracturas es un factor importante en el desarrollo estratégico de un yacimiento donde la permeabilidad está definida por la presencia de fracturas naturales.

Los pozos direccionales perforados perpendicularmente a la orientación de las fracturas muestran un incremento de las posibilidades de interceptar un mayor número de fracturas, y en consecuencia, de aumentar la productividad (Baker, 2008). En la Figura 1-3, se puede observar algunas fallas identificadas con el perfil de imagen sónico.

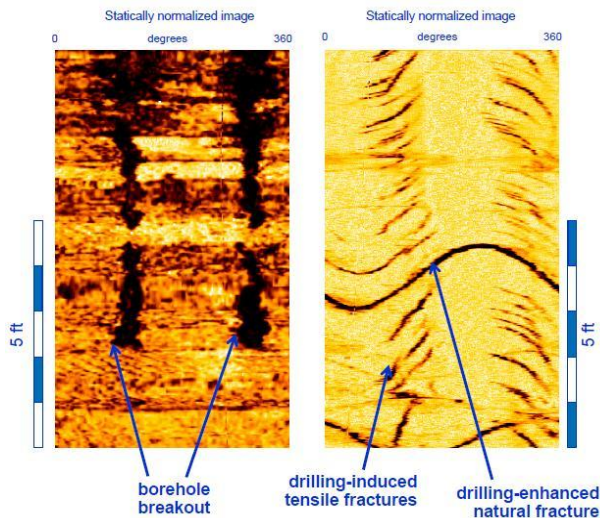


Figura 1-3. Perfil tomado con la herramienta CBIL de la compañía de servicios Baker Hughes (Baker, 2008).

o **Orientación de los esfuerzos**

Conocer la dirección de los esfuerzos horizontales (esfuerzo horizontal máximo y esfuerzo horizontal mínimo) tiene la misma importancia que conocer la orientación de las fracturas. Esto también puede tener un efecto positivo en la productividad. Cuando la dirección de la fractura es paralela a la dirección del esfuerzo mínimo, hay una tendencia de reducir la permeabilidad durante la producción, porque el esfuerzo horizontal principal tiende a cerrar la fractura durante y después del drenaje.

El ensanchamiento espontáneo del diámetro del pozo permite la determinación del estado de los esfuerzos en los alrededores de la cara del mismo. La información del ensanchamiento y de las fracturas inducidas ha sido usada con el objetivo de orientar la perforación de pozos muy desviados u horizontales, para reducir el riesgo del colapso del mismo durante la perforación y podría ser usado para el control del peso del lodo (Baker, 2008).

- o **Facies sedimentarias**

Las facies sedimentarias son el conjunto de sedimentos que pueden ser definidos y separados por su geometría, litología y estructura sedimentaria y fósiles asociados. En formaciones de arena, los perfiles sínicos de imagen son usados para determinar variaciones de las facies sedimentarias derivadas de diferentes niveles de compactación.

La variación en los colores dentro de la misma litología puede ser usada para definir las facies de acuerdo a la compactación (porosidad) con resoluciones verticales de menos de un pie, permitiendo que el tamaño del pozo permanezca relativamente constante.

Conocer el área y la relación entre los colores y la compactación de la arena, permite la perforación selectiva para la mejora de la productividad asociada con las capas más porosas y permeables. Los intervalos carbonatados también muestran intervalos más porosos con colores más oscuros por medio de la reducción de la amplitud sínica.

- **Perfil resistivo de Imagen**

El perfil resistivo de imagen es capturado por medio de herramientas eléctricas que generan una imagen de la formación a partir de mediciones de la resistividad de la zona invadida. Las herramientas en general están compuestas por patines articulados que consisten en una serie de sensores, los cuales se ponen en contacto con la pared del pozo y sobre ella emiten una corriente eléctrica que permite establecer una diferencia de potencial para determinar posteriormente la resistividad de la zona (Díaz, 2006).

A diferencia del perfil sínico de imagen, el perfil resistivo no puede obtener una vista de 360° del pozo, la amplitud de esta dependerá de la cantidad de patines con los que cuente la herramienta de perfilaje. La Figura 1-4, ilustra cómo la herramienta resistiva realiza el proceso de perfilaje.

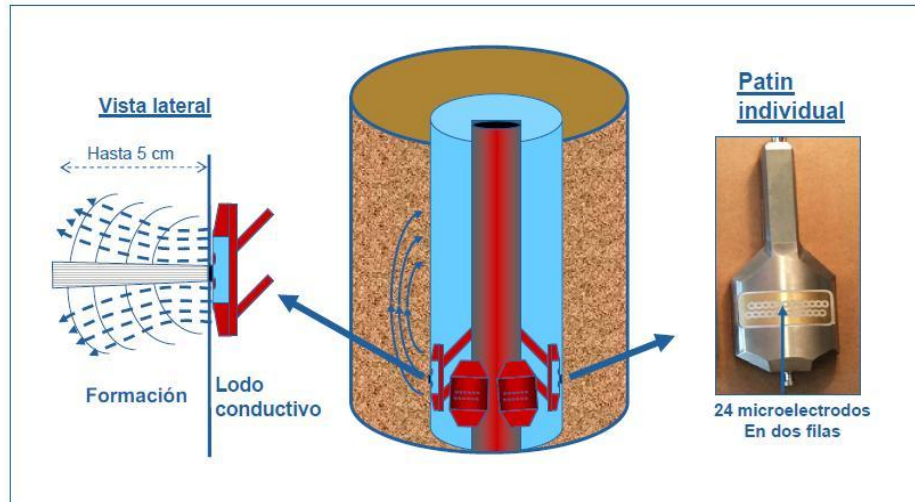


Figura 1-4. Toma de datos de con una herramienta resistiva (Díaz, 2006).

En la Figura 1-4 se puede apreciar cómo cada patín de la herramienta va aplicando corriente a la formación que lo rodea, mientras se va moviendo a lo largo del pozo girando sobre su propio eje. Por lo general todas las mediciones se toman de abajo hacia arriba, es decir, desde una profundidad determinada hasta la superficie del pozo.

Los perfiles resistivos de imagen son utilizados para determinar características geológicas de la pared del pozo entre las cuales se encuentran las siguientes:

- o Zonificación estructural (por análisis de buzamiento).
- o Interpretación de límites estructurales.
- o Integración del análisis de curvatura con los registros y datos sísmicos.
- o Caracterización de la fractura, la descripción de la fractura y su distribución.
- o Análisis del régimen de esfuerzos y parámetros geomecánicos.
- o Evaluación de la porosidad secundaria, determinación del espesor de arena neta.
- o Determinación de la dirección de las paleocorrientes.

La Figura 1-5 muestra un ejemplo de una imagen tomada con una herramienta de perfilaje resistiva.

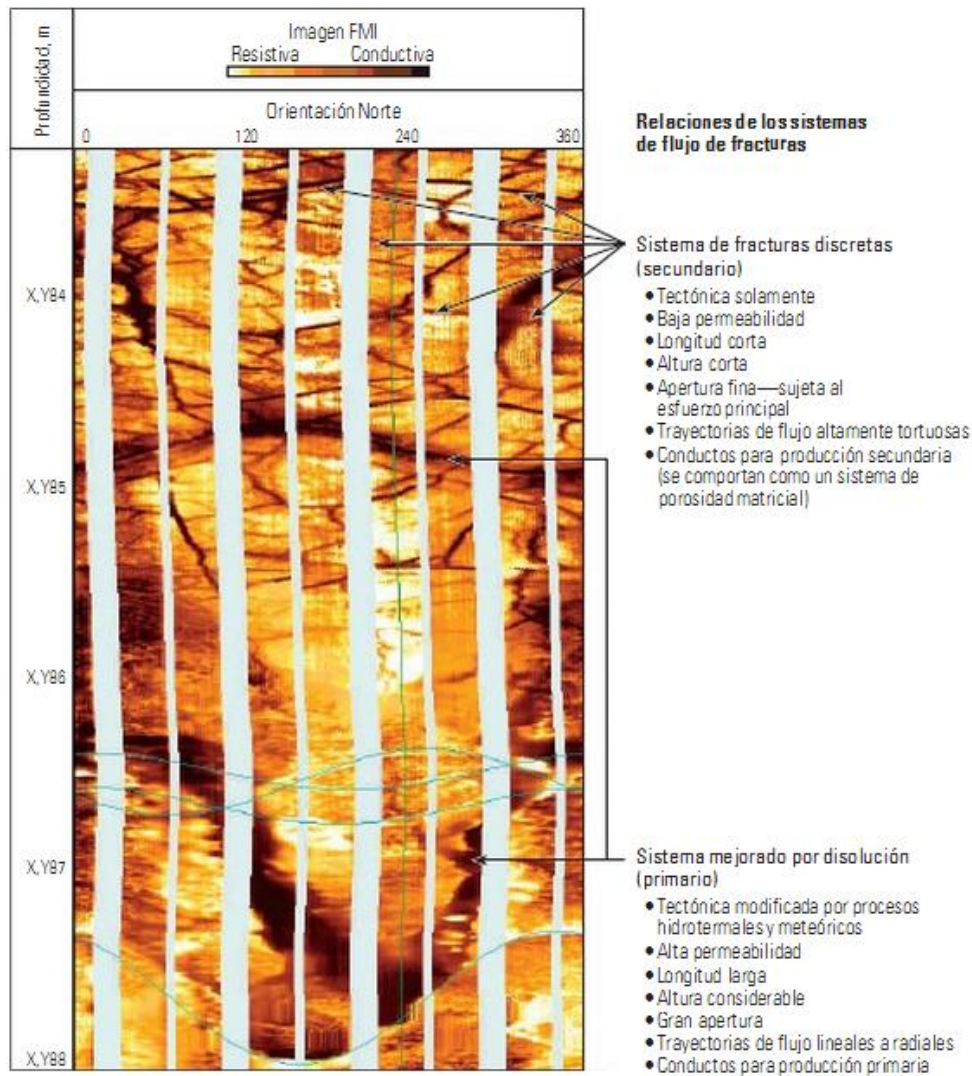


Figura 1-5. Registro de imagen con una herramienta resistiva (Díaz, 2006).

Los perfiles tomados por las diferentes herramientas (sónicas o resistivas) son almacenados en archivos que siguen estándares especializados para la industria de hidrocarburos, para luego ser interpretados por medio de un software que permite el análisis de los mismos. A continuación se describe uno de los estándares usados para este fin.

1.2 Estándar DLIS

El estándar DLIS, *Digital Log Interchange Standard*, fue la práctica recomendada número sesenta y seis (RP66) por el Instituto Americano del Petróleo en 1991. La Corporación Petrotécnica de Sistemas Abiertos o POSC, por sus siglas en inglés, adoptó el estándar DLIS en 1992, con el cual se fomentó su desarrollo como estándar sintáctico para sísmica, perforación y perfilaje de pozos. Actualmente este estándar cuenta con dos (2) versiones,

sin embargo esta investigación sólo describe la primera versión, ya que en este Trabajo Especial de Grado se trabaja con archivos DLIS V1.

El estándar DLIS propone un esquema de datos que permite almacenar, manejar e intercambiar datos de alta calidad. Este formato asegura la rastreabilidad requerida por la industria de Exploración y Producción (E&P), al especificar el equipamiento, las herramientas y los datos. Además, soporta un modo de clasificación de datos, con lo cual se facilita el acceso a estos. A continuación se describirá como se estructura los datos contenidos en el archivo DLIS.

1.2.1 Organización de los datos en un archivo DLIS

DLIS estructura los datos en un formato lógico que se encarga de la organización sintáctica y semántica de estos. En la Figura 1-6 se puede apreciar cómo se organizan los datos en un archivo DLIS.

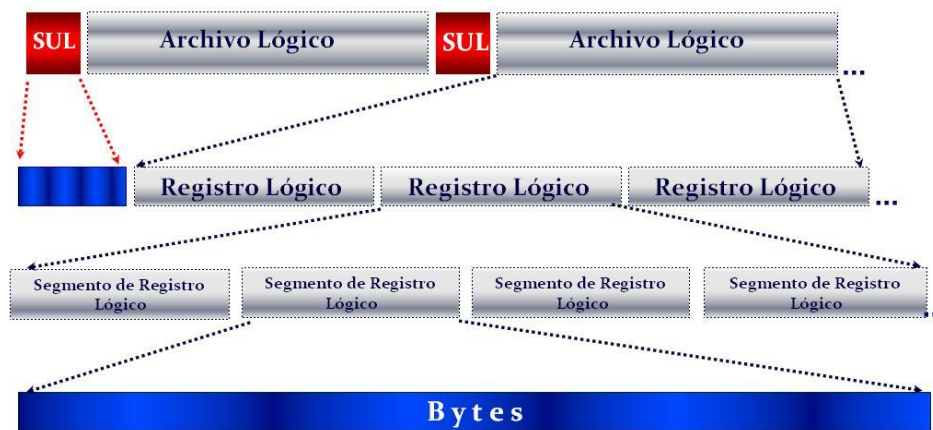


Figura 1-6. Organización de los datos en un DLIS (Chumbimuni, 2008).

A continuación se explicará cada uno de los componentes mostrados en la Figura 1-6.

- **Etiqueta de la unidad de almacenamiento**

La etiqueta de la unidad de almacenamiento o SUL, por sus siglas en inglés, está formada por 80 bytes de caracteres en código ASCII, los cuales están distribuidos como se muestra en la Tabla 1-2.

Campo	Tamaño en bytes
Número de secuencia de la unidad de almacenamiento	4
Versión del DLIS	5
Estructura de la unidad de almacenamiento	6
Tamaño máximo del registro	5
Identificador del conjunto de almacenamiento	60

Tabla 1-2. Distribución de la SUL (POSC, 1998).

- **Archivo lógico**

Un archivo lógico consiste en una secuencia de uno o varios registros lógicos, que comienzan con un *File Header Logical Record* (FHLR). Un Archivo Lógico termina cuando se encuentra con otro FHLR o cuando no existen más registros lógicos.

- **Registro lógico**

Los registros lógicos forman los cuerpos básicos coherentes de información en el estándar DLIS. Estos encapsulan la información semánticamente relacionada dentro de un Archivo Lógico. Cada Registro Lógico consiste en uno o varios Segmentos de Registros Lógicos consecutivos (Ver Figura 1-6).

Desde el punto de vista sintáctico, se tienen dos tipos de estructuras de registro lógico para el almacenamiento de información, los cuales son: registro lógico con formato explícito y registro lógico con formato indirecto. A continuación se explicará cada uno de estos:

- o **Registro lógico con formato explícito**

Un registro lógico con formato explícito o EFLR, por sus siglas en inglés, es una estructura que agrupa objetos del mismo tipo dentro de un conjunto. Un archivo DLIS contiene un EFLR por cada tipo de objeto. Cada EFLR contiene un sólo conjunto, el cual está formado por uno o más objetos del mismo tipo, estos objetos contienen uno o más atributos. Cabe destacar que los objetos son precedidos por una plantilla que se utiliza para especificar la presencia, orden y características de los objetos contenidos en el conjunto. En la Figura 1-7 se ilustra la estructura de un EFLR.

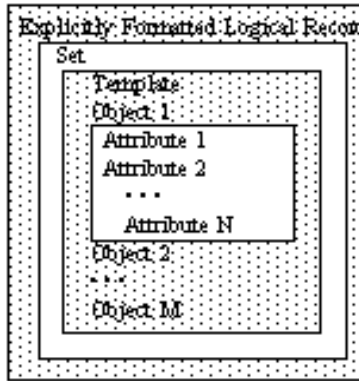


Figura 1-7. Estructura de un EFLR (POSC, 1998).

Para distinguir los conjuntos, atributos y objetos se utiliza el componente descriptor que se divide en dos (2) partes: el componente rol, que indica el tipo de entidad presente en el EFLR, los cuales pueden ser: atributo ausente, atributo, atributo invariante, objeto, conjunto redundante, conjunto de reemplazo y conjunto; y el componente formato, que indica las características del tipo de componente, la cuales pueden ser: nombre, código de representación, unidad, valor y tipo de objeto contenido en el conjunto.

- o **Registro lógico con formato indirecto**

El registro lógico con formato indirecto o IFLR, por sus siglas en inglés, es una estructura compuesta por: un descriptor de datos, que representa la relación entre un objeto contenido en un EFLR y el IFRL; y un formato de información indirecta, que es un conjunto de bytes sin formato que se extiende hasta el final de este registro lógico. El formato específico de este conjunto de bytes, es definido por el código de representación del objeto en el EFLR nombrado en la referencia del descriptor de datos. A través de este formato se podrá convertir el conjunto de bytes en información coherente que podrá ser utilizada para cálculos posteriores.

Como se indicó anteriormente el almacenamiento de datos del formato DLIS se hace mediante objetos, para esto se definen varios tipos, dependiendo de la información almacenada. Para el desarrollo del Trabajo Especial de Grado sólo se utilizarán tres (3) tipos de objetos los cuales serán definidos a continuación.

- **Objeto origen**

El objeto origen es aquel que contiene los datos generales sobre la muestra del perfil de pozo, indica quién creó los datos, cuándo, dónde y cómo fueron creados. Este objeto

contiene información relevante, tal como: nombre del archivo, nombre del pozo, herramientas utilizadas para la obtención de la muestra, fecha y hora en que se tomo la muestra, software que se utilizó para la creación del perfil del pozo, la versión del software, compañía de servicio que realizó el perfilaje, entre otros. Figura 1-8 muestra los atributos presentes en los objetos de tipo origen.

<i>Label</i>	<i>Restrictions</i>	<i>Label</i>	<i>Restrictions</i>
FILE-ID	C=1, R=ASCII	DESCENT-NUMBER	
FILE-SET-NAME	C=1, R=IDENT	RUN-NUMBER	
FILE-SET-NUMBER	C=1, R=UVARI	WELL-ID	C=1
FILE-NUMBER	C=1, R=UVARI	WELL-NAME	C=1, R=ASCII
FILE-TYPE	C=1, R=IDENT	FIELD-NAME	C=1, R=ASCII
PRODUCT	C=1, R=ASCII	PRODUCER-CODE	C=1, R=UNORN
VERSION	C=1, R=ASCII	PRODUCER-NAME	C=1, R=ASCII
PROGRAMS	R=ASCII	COMPANY	C=1, R=ASCII
CREATION-TIME	C=1, R=DTIME	NAME-SPACE-NAME	C=1, R=IDENT
ORDER-NUMBER	C=1, R=ASCII	NAME-SPACE-VERSION	C=1, R=UVARI

Figura 1-8. Atributos del objeto origen (POSC, 1998).

- **Objeto canal**

Los objetos canal es donde se almacena las muestras tomadas del pozo. Se presenta como una secuencia de datos indexados con respecto a: la profundidad, el tiempo o alguna otra dimensión física del pozo. Estos datos vienen expresados como una matriz, de tamaño Mx1 en el caso de las curvas y MxN en caso de las imágenes. Cabe destacar que la matriz de datos del objeto canal son almacenados de forma indirecta, por lo que para acceder a estos datos se hace mediante el IFLR. Adicional a esto el objeto canal cuenta con atributos mostrados en la Figura 1-9 los cuales estarán presentes en el EFLR de este tipo de objetos.

Label	Restrictions
LONG-NAME	C=1, R=(OBNAME< ASCII)
PROPERTIES	R=IDENT
REPRESENTATION-CODE	C=1, R=USHORT
UNITS	C=1, R=UNITS
DIMENSION	R=UVARI
AXIS	R=OBNAME
ELEMENT-LIMIT	R=UVARI
SOURCE	C=1, R=OBJREF

Figura 1-9. Atributos del objeto canal (POSC, 1998).

- **Objeto trama**

Cada objeto trama define un tipo de estructura donde se enumera y se hace referencia los canales grabados, se define el rango de profundidad y la dirección (ascendente o descendente) en que fue tomada la muestra. En la Figura 1-10 se muestra los atributos que contiene el objeto trama.

Label	Restrictions
DESCRIPTION	C=1, R=ASCII
CHANNELS	R=OBNAME
INDEX-TYPE	C=1, R=IDENT
DIRECTION	C=1, R=IDENT
SPACING	C=1
ENCRYPTED	C=1, R=USHORT
INDEX-MIN	C=1
INDEX-MAX	C=1

Figura 1-10. Atributos del objeto trama (POSC, 1998).

El principal objetivo de este capítulo fue describir algunos de los fundamentos teóricos necesarios para entender la situación problemática a resolver con el desarrollo de este Trabajo Especial de Grado. En tal sentido, se puede concluir que el perfilaje de pozos es una técnica importante para la industria petrolera, específicamente en el área de exploración y producción, debido a que por medio del perfilaje se obtiene información relevante que reduce la incertidumbre sobre las propiedades del pozo. La información proveniente del

perfilaje es almacenada en un archivo de tipo DLIS, por esta razón es imprescindible contar con una herramienta que procese el archivo y muestre e intérprete los datos contenidos en ellos.

A continuación, el Capítulo 2 describe el desarrollo del producto de software resultado de este Trabajo Especial de Grado aplicando el modelo de procesos MeRinde.

Capítulo 2

Desarrollo de la aplicación VenLog 2.0 siguiendo el modelo de procesos MeRinde

En este capítulo se define el modelo de procesos MeRinde, su estructura y las disciplinas requerimientos, análisis y diseño, implementación y pruebas, pertenecientes a este modelo de procesos, que se desarrollaron durante la realización de este Trabajo Especial de Grado.

2.1 Modelo de proceso MeRinde

MeRinde (Metodología de la Red Nacional de Integración y Desarrollo de Software Libre), es un modelo de proceso definido por los miembros del Centro Nacional de Tecnologías de Información (CNTI), que propone un estándar abierto para el proceso de desarrollo de software orientado a planes (Marrero & Santos, 2007). El desarrollo orientado a planes, también conocido como metodologías pesadas, esta orientado al control de procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán (Gabardini, 2004).

El modelo MeRinde surge para dar respuesta a la necesidad del Estado venezolano de gestionar proyectos de Software Libre, cumpliendo con los estándares internacionales que propician un software de calidad (Marrero & Santos, 2007). Es por esto, que este modelo ha sido adoptado por la Fábrica de Software de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT) localidad Miranda Oeste de PDVSA, como estándar en los proyectos de desarrollo de aplicaciones especializadas de software para el negocio y sus filiales, enmarcados en el Plan Siembra Petrolera de PDVSA.

MeRinde establece una estructura que cubre todo el ciclo de vida del software, por esto incluye fases, roles, actividades, artefactos, disciplinas, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración. Cabe destacar que este modelo es iterativo e incremental, ya que continuamente estará refinando, corrigiendo o mejorando los artefactos del sistema. A continuación se describe la estructura de este modelo de procesos.

2.2 Estructura de MeRinde

MeRinde propone un estándar abierto para el proceso de desarrollo de software orientado a planes que se estructura en dos (2) dimensiones o ejes (eje vertical y eje horizontal) como se muestra en la Figura 2-1. Cada barra representa una iteración por fase para un proyecto, adicionalmente el modelo enfatiza que durante cada iteración se recorren todas las disciplinas pero con diferente esfuerzo.

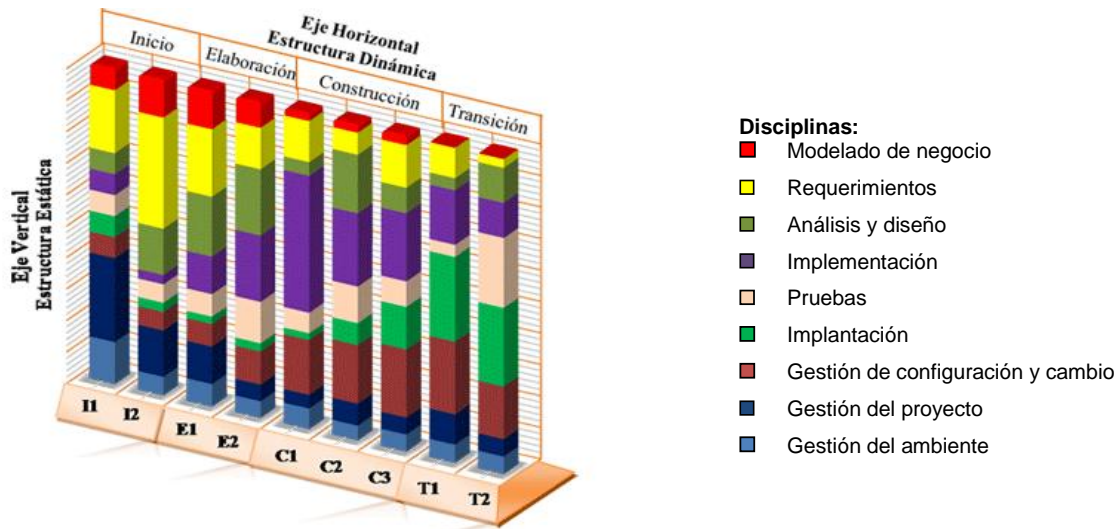


Figura 2-1. Esfuerzo en actividades según las fases de proyecto de MeRinde (Marrero & Santos, 2007).

Como se muestra en la Figura 2-1, se tienen dos ejes y consisten en lo siguiente:

- Eje horizontal: representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso, expresado en términos de fases, iteraciones e hitos.
- Eje vertical: representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes, disciplinas, actividades, artefactos y roles.

Dado que el aporte suministrado en este Trabajo Especial de Grado, forma parte de un módulo en particular del proyecto *Suite de Perforación* que se está llevando a cabo por la Gerencia de Construcción de Pozos de PDVSA Intevep en conjunto con la Gerencia de AIT localidad Miranda Oeste de PDVSA, este proyecto no abarca todas las disciplinas mostradas en la Figura 2-1, sólo se enfoca en las disciplinas: requerimientos, análisis y diseño, implementación y pruebas de software, generando sólo los artefactos necesarios de cada una de éstas. Las disciplinas que corresponden a la gestión del proyecto (configuración de cambios, gestión de proyecto y gestión del ambiente) son llevadas por el

líder del proyecto *Suite de Perforación*, y para el caso de la implantación corresponde al equipo de mantenimiento de la Gerencia de AIT localidad Miranda Oeste de PDVSA. A continuación, se presentan las disciplinas que abarca el Trabajo Especial de Grado.

2.2.1 Requerimientos

Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario. Los requerimientos pueden ser divididos en dos grupos: Los requerimientos funcionales, los cuales describen las funciones que el software va a ejecutar; por ejemplo, ajustarse a un formato de texto o modular una señal, y los requerimientos no funcionales, los cuales especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus funciones específicas; por ejemplo, seguridad, eficiencia y usabilidad. A continuación se describen los requerimientos no funcionales y funcionales de la herramienta VenLog 2.0.

2.2.1.1 Requerimientos no funcionales

Para el desarrollo de VenLog 2.0, durante el levantamiento de los requerimientos, se plantearon los siguientes requerimientos no funcionales:

- **Usabilidad:** El software será de fácil uso, ya que las interfaces estarán diseñadas pensando en las necesidades del usuario. El software contará con iconos alusivos y textos de ayuda que faciliten la interacción con el programa, por lo que el tiempo de aprendizaje será mínimo.
- **Confiabilidad:** El software será confiable, ya que, asegura un nivel de funcionamiento adecuado bajo condiciones normales, y se recupera manteniendo su nivel de funcionamiento en caso de errores.
- **Mantenimiento y actualización:** El sistema será construido de forma modular, con alta cohesión y bajo acoplamiento entre módulos, por lo que el mantenimiento y actualización será una tarea sencilla ya que los cambios realizados en un módulo no afectaran a otro.
- **Soportabilidad y operabilidad:** El sistema cumplirá con soportabilidad ya que este puede ser proporcionado por el grupo de desarrolladores de la *Suite de Perforación*. Será operable debido a que el software permitirá al usuario operarlo y controlarlo de una manera sencilla e intuitiva.

2.2.1.2 Requerimientos funcionales

Para el desarrollo de VenLog 2.0, durante el levantamiento de los requerimientos, se plantearon los siguientes requerimientos funcionales:

- **Interpretar archivos DLIS:** Es la funcionalidad que permite interpretar los datos contenidos en un archivo DLIS y mostrar al usuario los datos más relevantes.
- **Construir curva:** Es la funcionalidad que permite construir una curva a partir de una matriz de datos de tamaño $M \times 1$.
- **Construir imagen:** Es la funcionalidad que permite construir una imagen a partir de una matriz de datos de tamaño $M \times N$.
- **Construir imagen resistiva:** Es la funcionalidad que permite construir una imagen resistiva a partir de la curva de azimut y cuatro, seis u ocho canales de imágenes equivalentes a los patines⁴ de la herramienta de perfilaje resistivo.
- **Cambiar color:** Es la funcionalidad que permite cambiar de color a las curvas generadas.
- **Cambiar paleta:** Es la funcionalidad que permite cambiar la tonalidad de las imágenes generadas.
- **Zoom:** Esta funcionalidad permite aumentar o reducir el 25% de la escala vertical actual de la imagen o curva. También permite restaurar la escala vertical, de la imagen o curva, a su escala original.
- **Ver cuadrícula:** Es la funcionalidad que permite mostrar u ocultar una cuadrícula en las curvas generadas.

En la Figura 2-2 se muestra el diagrama general de casos de uso de VenLog 2.0.

⁴ Patín(es): es el objeto con el que la herramienta de perfilaje resistiva toma sus muestras. Las herramientas de perfilaje resistiva pueden tener cuatro, seis u ocho patines.

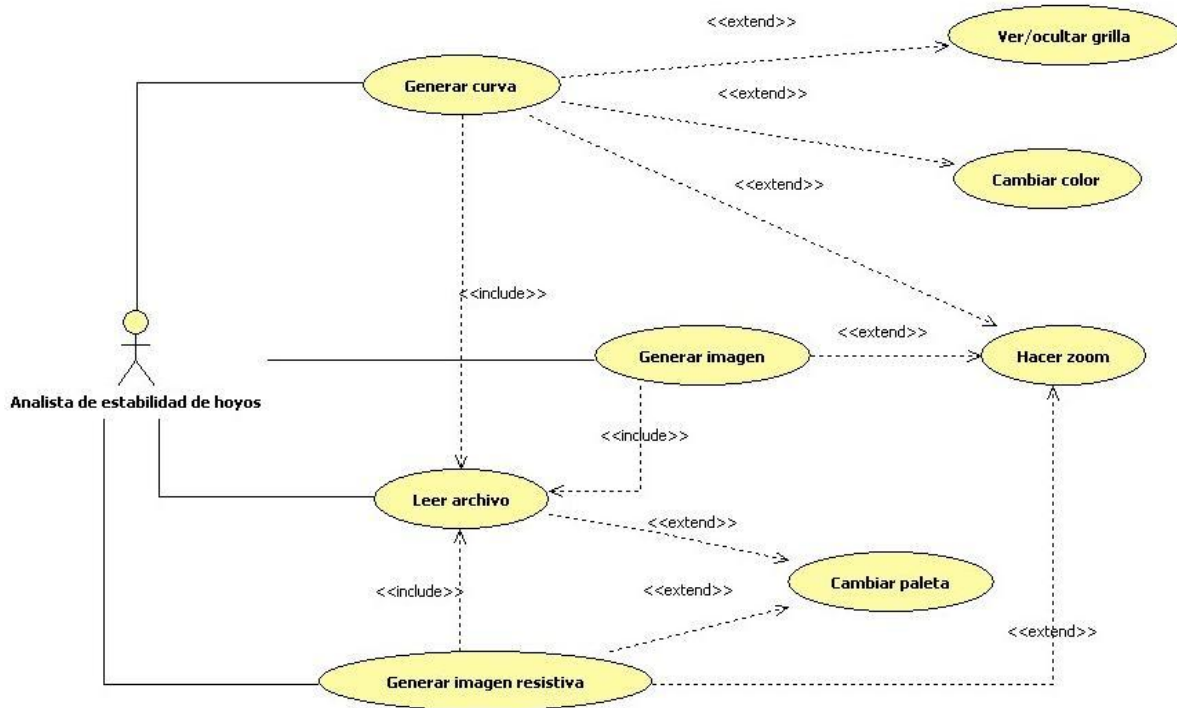


Figura 2-2. Diagrama de casos de uso de VenLog 2.0 (módulo lector de registros).

A continuación en la Tabla 2-1 se presenta la descripción o especificación del caso de uso Leer archivo.

Caso de Uso-001	
Nombre:	Leer archivo
Descripción:	Es la funcionalidad que permite interpretar un archivo DLIS y mostrar los datos relevantes para el usuario.
Requerimiento:	Interpretar DLIS
Precondición:	
Flujo Normal:	
Actor	Sistema
1. Introduce la ruta donde se encuentra el archivo DLIS.	2. Valida la ruta 3. Verifica que el archivo tenga una estructura valida. 4. Realiza la interpretación 5. Muestra los datos al usuario.
Flujo Alternativo 1:	
Actor	Sistema

Caso de Uso-001	
1.1. Introduce la ruta donde se encuentra el archivo DLIS.	1.2 Valida la ruta 1.3 Manda mensaje de ruta inválida.
Flujo Alterno 2:	
Actor	Sistema
2.1 Introduce la ruta donde se encuentra el archivo DLIS.	2.2 Valida la ruta. 2.3 Verifica que el archivo tenga una estructura valida. 2.4 Manda mensaje de archivo dañado.
Postcondición:	
Requerimientos Especiales:	Adicionalmente esta funcionalidad permite exportar el archivo a formato HTML para que pueda ser legible para el usuario.
Puntos de Extensión:	
Puntos de Inclusión	<ul style="list-style-type: none"> • Generar imagen • Generar imagen resistiva • Generar curva

Tabla 2-1. Descripción caso de uso Leer archivo.

A continuación en la Tabla 2-2 se presenta la descripción o especificación del caso de uso Generar curva.

Caso de Uso-002	
Nombre:	Generar curva
Descripción:	Es la funcionalidad que permite construir una curva a partir de una matriz de datos de tamaño Mx1.
Requerimiento:	Construir curva
Precondición:	Haber ejecutado el CU Leer archivo
Flujo Normal:	
Actor	Sistema
<ol style="list-style-type: none"> 1. Selecciona el (los) canal(es) que desea graficar 2. Envía la petición al sistema 	<ol style="list-style-type: none"> 3. Valida que el tamaño de la(s) matriz(es) de datos sea(n) de Mx1 4. Traduce el conjunto de bytes al tipo de dato asociado al (los) canal(es). 5. Construye la(s) curva(s). 6. Muestra la(s) curva(s) al usuario.
Flujo Alterno 1:	
Actor	Sistema
1.1. Selecciona el(los) canal(es) que desea graficar.	1.3 Valida que el tamaño de la(s) matriz(es) de datos sea(n) de Mx1.

Caso de Uso-002	
1.2 Envía la petición al sistema	1.4 Manda mensaje de tamaño inválido.
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	<ul style="list-style-type: none"> • Cambiar color • Ver/Ocultar grilla • Hacer zoom
Puntos de Inclusión	

Tabla 2-2. Descripción caso de uso Generar curva.

A continuación en la Tabla 2-3 se presenta la descripción o especificación del caso de uso Generar imagen.

Caso de Uso-003	
Nombre:	Generar imagen
Descripción:	Es la funcionalidad que permite construir una imagen a partir de una matriz de datos de tamaño MxN.
Requerimiento:	Construir imagen
Precondición:	Haber ejecutado el CU Leer archivo
Flujo Normal:	
Actor	Sistema
<ol style="list-style-type: none"> 1. Selecciona el(los) canal(es) que desea graficar. 2. Envía la petición al sistema 	<ol style="list-style-type: none"> 3. Valida que el tamaño de la(s) matriz(es) de datos sea(n) de MxN 4. Traduce el conjunto de bytes al tipo de dato asociado al (los) canal(es). 5. Construye la(s) imagen(es). 6. Muestra la(s) imagen(es) al usuario.
Flujo Alternativo 1:	
Actor	Sistema
<ol style="list-style-type: none"> 2.1. Selecciona el(los) canal(es) que desea graficar. 2.2 Envía la petición al sistema 	<ol style="list-style-type: none"> 2.3 Valida que el tamaño de la(s) matriz(es) de datos sea(n) de MxN. 2.4 Manda mensaje de tamaño inválido.
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	<ul style="list-style-type: none"> • Cambiar paleta • Hacer zoom
Puntos de Inclusión	

Tabla 2-3. Descripción caso de uso Generar imagen.

La descripción de los demás casos de uso se encuentra en el documento de Especificación de Requerimientos de Software (ver Anexo 1), el cual es el artefacto que se desarrolló durante el progreso de la disciplina requerimientos. El objetivo de este artefacto es describir las funciones del sistema, los requerimientos no funcionales, características de diseño y prototipos de las interfaces de usuario, con el fin de proporcionar una descripción completa y comprensiva de los requerimientos para el software a desarrollar. A continuación se explica la disciplina análisis y diseño.

2.2.2 Análisis y diseño

El objetivo principal de la disciplina análisis y diseño es transformar los requerimientos a una especificación que describa cómo implementar el sistema. El análisis fundamentalmente consiste en ver qué hace el sistema de software a desarrollar, por tal motivo se enfoca en los requerimientos funcionales. Por otro lado el diseño es un refinamiento que abarca los requerimientos no funcionales, por lo cual se centra en cómo el sistema cumple sus objetivos.

Durante el progreso de esta disciplina se definió y diseñó la arquitectura, la cual consiste en dos (2) capas que separa los datos de la aplicación en presentación y lógica de control. La capa de presentación es la encargada de la navegabilidad del sistema, la presentación de los datos, interfaz gráfica de usuario y la validación de los datos de entrada; mientras que la capa lógica de control es la encargada de hacer la interpretación de los archivos DLIS. La Figura 2-3 muestra el diagrama de componentes de VenLog 2.0 dividido en capas.

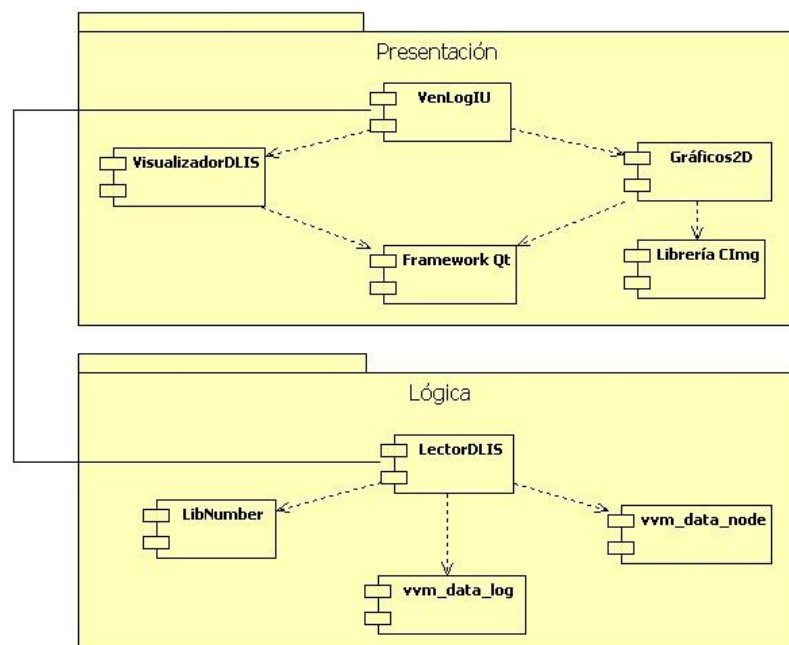


Figura 2-3. Diagrama de componentes VenLog 2.0.

Además de definir cómo sería la arquitectura, se definió las estructuras de las clases que componen cada una de las capas mostradas en la Figura 2-3. A continuación en la Figura 2-4 se muestra parte del diagrama de clases correspondiente a la capa de presentación. El diagrama completo se encuentra en el Anexo 2.

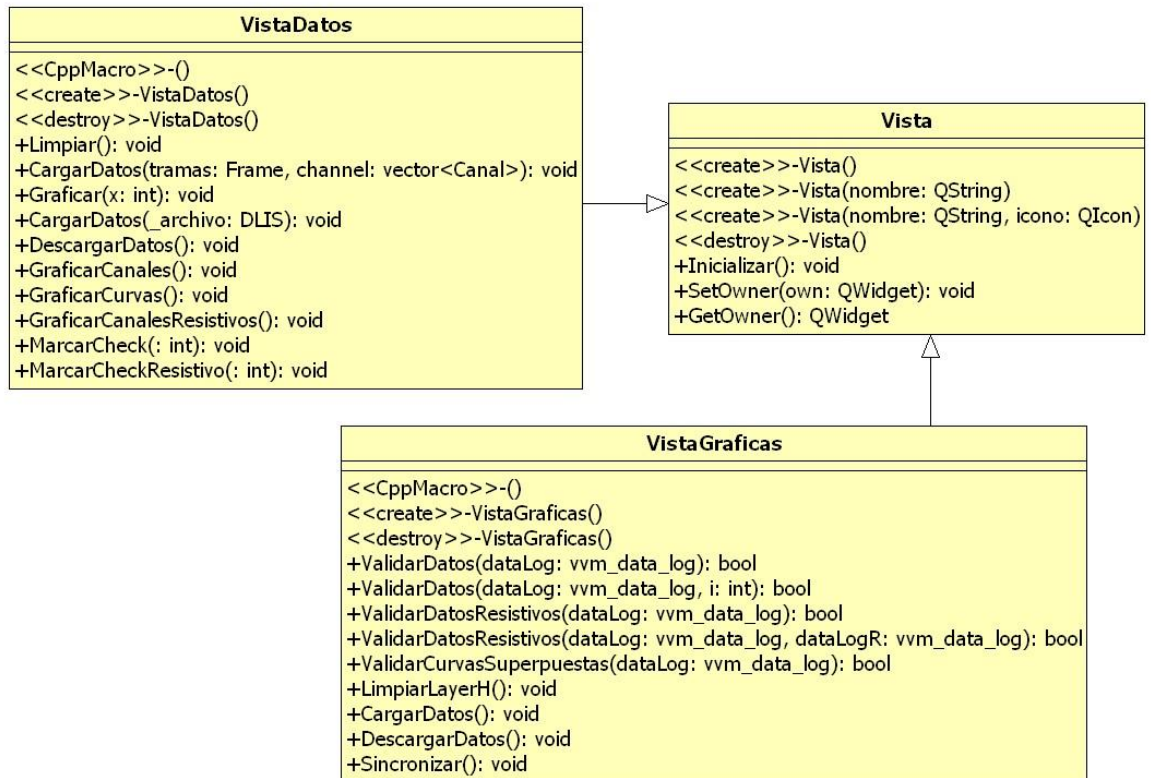


Figura 2-4. Diagrama de clase capa presentación VenLog 2.0.

Durante el desarrollo de esta disciplina se generó el artefacto Documento de Arquitectura del Software (ver Anexo 2), el cual es una especificación de las ideas principales del diseño. Éste proporciona una descripción de la arquitectura del sistema software, y sirve como medio de comunicación entre el arquitecto de software y otros miembros del equipo de proyecto, con respecto a las decisiones arquitectónicamente significativas que se han tomado en el mismo. Contiene varias vistas que muestran aspectos distintos del sistema como son: Vista de Casos de Uso (diagrama de casos de uso), Vista Lógica (diagrama de clases), Vista de Implementación (diagrama de componentes), entre otros datos relevantes como aspectos de calidad e integración de componentes.

2.2.3 Implementación

El objetivo principal de la disciplina implementación es convertir los elementos del diseño en elementos de implementación, estos elementos son: códigos fuentes, ejecutables y

binarios. Otra parte de esta disciplina son las pruebas unitarias, las cuales se limitan a los componentes de software implementados. De esta disciplina se obtiene un sistema ejecutable estable, constituido de los resultados producidos por los programadores individuales.

Debido a que este proyecto fue desarrollado para PDVSA Intevep siguiendo los lineamientos de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT) localidad Miranda Oeste, se utilizaron las siguientes herramientas de software: C++ como lenguaje de programación base, el framework Qt para la creación de interfaces de usuario, CImg como librerías para la generación de gráficas, el compilador MinGW y el entorno de desarrollo integrado (IDE) Qt Creator, para facilitar el desarrollo del código fuente. Dado que la plataforma de escritorio actual de PDVSA Intevep es Windows XP, estas herramientas se ejecutaron bajo este sistema operativo. En la Figura 2-5 se ilustra cómo está constituida la plataforma de desarrollo para este Trabajo Especial de Grado.



Figura 2-5. Estructura de la plataforma de desarrollo.

A continuación se explican las herramientas de software indicadas anteriormente.

Sistema Operativo Windows XP

El sistema operativo Windows XP es un software desarrollado por Microsoft que se encarga de administrar los recursos físicos (procesador, memoria, unidades de almacenamiento, entre otros) de manera eficiente, confiable, tolerando y enmascarando las fallas. Windows XP también se encarga de proveer un ambiente de ejecución para las aplicaciones corren sobre el computador (programas de ofimática, de desarrollo, de diseño, entre otros).

A pesar que el decreto 3390 (ver Anexo 3) dispone que: “*la Administración Pública Nacional (APN) empleará prioritariamente Software Libre desarrollado con estándares*”

abiertos, en sus sistemas, proyectos y servicios informáticos”, la plataforma de escritorio de PDVSA Intevep usa como sistema operativo Windows XP, ya que los investigadores de esta filial requieren hacer uso de licencias de software especializado que sólo corren sobre este sistema operativo. Es por esto, que el producto final de este Trabajo Especial de Grado, fue desarrollado bajo Windows XP, aunque se podrá ejecutar en sistemas GNU/Linux por requerimientos de la empresa ya que se tiene planeado, por parte de estos, la migración progresiva de la plataforma.

Compilador MinGW

MinGW (Minimalist GNU for Windows), anteriormente conocido como MinGW32, es una implementación de los compiladores GCC para la plataforma Win32, que permite migrar la capacidad de este compilador en entornos Windows. MinGW incluye un conjunto de la API de Win32, permitiendo un desarrollo de aplicaciones nativas para esa plataforma, pudiendo generar ejecutables y bibliotecas usando la API de Windows. Para lenguajes de programación distintos de C, MinGW, utiliza las bibliotecas de tiempo de ejecución GNU (por ejemplo, GNU libstdc++ para C++) (MinGW.org, 2012).

Al utilizar este compilador el sistema creado podrá ser ejecutado tanto en plataformas Windows como en GNU/Linux sin necesidad de realizar cambios relevantes en el código, esto garantiza que el producto final seguirá siendo usado por la empresa al migrar la plataforma.

Lenguaje de programación C++

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que C++ es un lenguaje de programación multiparadigma. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores modernos. Una particularidad de C++ es la posibilidad de redefinir los operadores, y de poder crear nuevos tipos que se comporten como tipos fundamentales (Amaya, 2012).

El lenguaje de programación C++ presenta las siguientes características:

- Es un lenguaje abierto que posee un estándar ISO, durante un tiempo, C++ no tenía ninguna norma oficial y se mantuvo por un estándar de facto, sin embargo desde el año 1998, C++ está estandarizado por un comité de la ISO.
- Es un lenguaje compilado, ya que compila directamente al código nativo de la máquina, lo que le permite ser uno de los lenguajes más rápidos.
- Es un lenguaje fuertemente tipado, debido a que no permite violaciones de los tipos de datos, es decir, dado una variable de un tipo concreto, no se puede usar como si fuera una variable de otro tipo distinto a menos que se haga una conversión.
- Es compatible con la comprobación de tipo estático y dinámico, por lo que permite conversiones de tipos a verificar en tiempo de compilación o en tiempo de ejecución, por lo que ofrece otro grado de flexibilidad.
- Es portable, ya que tiene una amplia gama de compiladores que se ejecutan en diferentes plataformas que lo soportan. El código escrito en C++ que usa solo la biblioteca estándar se puede ejecutar en varias plataformas con pocos o ningún cambio.

Haciendo uso de este lenguaje se implementó el producto final del Trabajo Especial de Grado.

Framework Qt

El framework Qt es usado para desarrollar aplicaciones con una interfaz gráfica de usuario, así como también para el desarrollo de programas sin interfaz gráfica como herramientas para líneas de comandos y consolas para servidores. Este framework es multiplataforma y es utilizado en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros. Qt utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado en otros lenguajes de programación a través de *bindings*⁵.

Qt funciona en las principales plataformas (Windows, Linux y Mac OS) y tiene un amplio apoyo por la comunidad de Software Libre. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos, además de estructuras de datos tradicionales.

Este framework se utilizó para la creación de la interfaz gráfica de usuario, no fue necesario el uso de *bindings* ya que el programa fue escrito en el lenguaje de programación nativo de

⁵ binding es una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquél en el que ha sido escrita.

Qt (C++), y al ser multiplataforma podrá ser ejecutado en Windows y en distribuciones GNU/Linux por lo cual se adapta correctamente a las necesidades de la empresa.

Librería CImg

CImg es una librería, de código abierto escrita en C++ la cual provee un kit de herramientas para el procesamiento de imágenes, diseñada con las siguientes características (Tschumperlé, 2004):

- **Usabilidad:** CImg define las clases y métodos para administrar las imágenes en el mismo código C++. Se puede utilizar para cargar y guardar gráficos en varios formatos de archivos, acceder a los valores de píxel, mostrar y transformar imágenes de filtros, dibujar primitivas (texto, caras, curvas, objetos 3D, entre otros), realizar estadísticas de cálculo y gestionar las interacciones del usuario con las imágenes de manera sencilla.
- **Generalidad:** CImg define una clase única de imagen que puede representar un conjunto de datos que tiene hasta 4-dimensiones, a partir de señales escalares 1D (imágenes hiperespectrales) a 3D (imágenes volumétricas), construyendo los píxeles con distintos tipos de datos (bool, char, float, int, entre otros).
- **Portabilidad:** CImg es autónomo y por lo tanto altamente portátil. Trabaja con distintos sistemas operativos (Unix, Windows, MacOS X) y es compatible con varios compiladores C++ (Visual C++, g++, clang++, ICC).
- **Simplicidad:** CImg se compone de una cabecera de archivo único CImg.h que se debe incluir en un archivo fuente de C++. Se define sólo cuatro clases diferentes (CImg<T>, CImgList<T>, CImgDisplay y CImgException) encapsulados en un espacio de nombres. Puede ser compilado con un conjunto mínimo de estándares de C++ y las bibliotecas del sistema.
- **Extensibilidad:** Aunque no es obligatorio, CImg puede utilizar las funciones externas de otras librerías tales como: ImageMagick, GraphicsMagick, XMedCon, FFMPEG, libpng, libjpeg, libtiff, Magick++, OpenCV, Lapack, Board, OpenEXR o FFTW3.
- **Código abierto:** CImg es una librería de código abierto distribuida bajo la licencia CeCILL-C o CeCILL, la cual puede ser usada en aplicaciones comerciales.

Haciendo uso de esta librería fue posible generar las imágenes y curvas asociadas al perfil del pozo. De este modo permitirle al usuario realizar los estudios y análisis correspondientes a cada tipo de gráfica.

Entorno de desarrollo integrado Qt Creator

Qt Creator es un entorno de desarrollo integrado (IDE) que proporciona las herramientas necesarias para diseñar y desarrollar aplicaciones con el framework Qt. Los sistemas operativos que lo soportan son:

GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para Linux con gcc 3.3.

Mac OS X 10.4 o superior, requiriendo Qt 4.x

Windows XP y superiores, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW (Corporation, 2011).

Qt Creator sirvió de ayuda en la creación de VenLog 2.0, ya que este IDE permitió la escritura, compilación y ejecución del código fuente. En la Figura 2-6 se puede apreciar el uso de las herramientas de software descritas anteriormente en la disciplina implementación haciendo uso del Plugin para StarUML del estándar SPEM.⁶

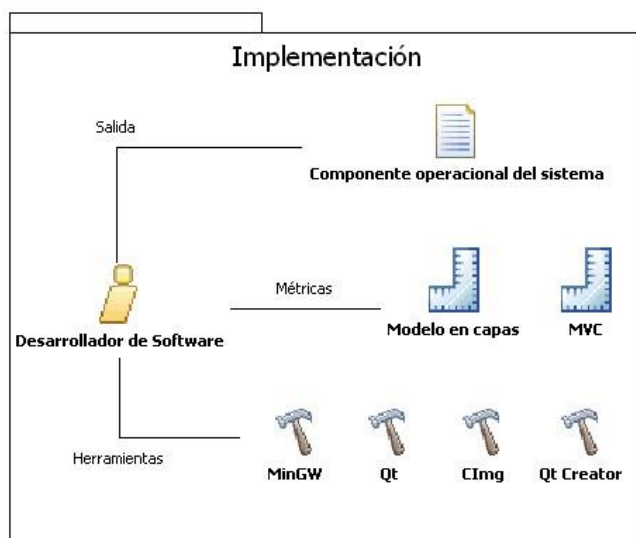


Figura 2-6. Integración del modelo de proceso con las herramientas de software en la disciplina implementación.

⁶ SPEM (Software Poces Engineering Metamodel), es un perfil de UML basado en MOF, que permite definir concretamente procesos o familias de procesos de software.

MOF (MetaObject Facility), es un lenguaje para definir lenguajes de modelado a partir de metamodelos. Fue definido por la OMG y actualmente se encuentra en su versión 2.0 [OMG 2010]. El objetivo de MOF es la especificación y gestión de metadatos en diferentes niveles de abstracción.

Perfil UML, en inglés profile UML constituyen el mecanismo que proporciona el propio UML para extender su sintaxis y su semántica para expresar los conceptos específicos de un determinado dominio de aplicación [Wiki 2010].

2.2.4 Pruebas

El desarrollo de esta disciplina consiste en planificar lo que hay que probar, diseñar cómo se llevarán a cabo las pruebas, implementar lo necesario para llevarlas a cabo, ejecutarlas en los niveles necesarios y obtener los resultados, de forma que la información obtenida sirva para ir mejorando el producto a desarrollar. El papel de las pruebas no es asegurar la calidad, pero sí evaluarla, y proporcionar una realimentación a tiempo, de forma que los aspectos de calidad puedan resolverse de manera efectiva en tiempo y costo.

Para el desarrollo de esta disciplina se realizaron tres (3) tipos de pruebas, a saber: pruebas de integración, pruebas de sistema y pruebas de aceptación. Estas pruebas fueron ejecutadas bajo la configuración de dos (2) ambientes de pruebas y un conjunto de herramientas, tanto de software como de hardware. Las herramientas utilizadas así como la configuración de los ambientes de prueba se muestran en las tablas a continuación.

En la Tabla 2-4 se especifican las herramientas de software utilizadas para los distintos tipos de pruebas.

Nombre	Versión	Descripción	Tipo de Prueba
Sistema Operativo	2002	Windows XP professional, service pack 3	Todas
VenLog	2.0	Aplicación propia, escrita en C++ con el Framework Qt 4.7	Todas
Qt Creator	2.2.1	IDE del Framework Qt	Sistema e integración
MinGW	3.81	Compilador de C++ basado en GCC para plataforma Win32	Sistema e integración

Tabla 2-4. Herramientas de software para la ejecución de las pruebas.

En la Tabla 2-5 se especifican las herramientas de hardware utilizadas para los distintos tipos de pruebas.

Recurso	Cantidad	Descripción
Procesador	1	AMD Sempron (tm) 130 2,61GHz
RAM	1	896 mb
Disco duro	2	1. Capacidad 30GB y disponible 19GB 2. Capacidad 430GB y disponible 376GB

Tabla 2-5. Herramientas de hardware para la ejecución de las pruebas.

En la Tabla 2-6 se muestran los dos (2) ambientes de configuración que se utilizaron en las pruebas realizadas.

Nombre de Configuración	Descripción	Implementación de la Configuración Física
Estándar	Configuración estándar para realizar las pruebas fuera del ambiente de desarrollo.	Instalador de la aplicación
Nativa	Configuración específica para pruebas dentro del ambiente de desarrollo	Compilador MinGw ID Qt Creator

Tabla 2-6. Configuración del ambiente de pruebas.

A continuación se describen los distintos casos de pruebas ejecutados con sus respectivos resultados.

Pruebas de integración

Las pruebas de integración son realizadas por los programadores del sistema en el ambiente de desarrollo. La Tabla 2-7 describe en qué consisten las pruebas de integración.

Objetivo:	Verificar la correcta interacción entre los componentes del sistema
Descripción:	Las pruebas de integración verifican si los componentes o subsistemas interactúan correctamente a través de sus interfaces, cubren la funcionalidad establecida, y se ajustan a los requerimientos especificados-
Técnicas:	<p><i>Pruebas de integración estructural:</i> estas pruebas son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirse a sentencias del lenguaje, se refieren a llamadas entre módulos</p> <p><i>Pruebas de integración funcional:</i> estas pruebas son similares a las pruebas de caja negra. Aquí se trata de encontrar fallos en la respuesta de un módulo cuando su operación depende de los procedimientos o funciones de otro(s) módulo(s).</p>
Fases:	Integrar la IU con el módulo lector y a partir de esta ir integrando los demás módulos.
Entorno de prueba:	Se realizarán pruebas que verifiquen la integración de todos los módulos del sistema, usando diferentes archivos de entrada.
Criterios de Éxito:	Los módulos funcionan correctamente de manera conjunta

Tabla 2-7. Pruebas de integración.

La Tabla 2-8 muestra en detalle las pruebas de integración realizadas a VenLog 2.0 con sus respectivos resultados.

ID/Nombre/Sistema/Proyecto: VenLog 2.0	Nivel de Prueba: Parcial
ID Caso de Uso:	Tipo(s) de Pruebas(s): de integración
ID Requerimiento:	Ambiente de Prueba: Nativa

ID/Nombre Escenario:		Autor del Caso de Prueba: Darwin Martínez		
ID/Nombre Caso de Prueba: CP-01		Nombre del Probador:		
Versión del Caso de Prueba: 1		Fecha de Creación: 31/05/12	Fecha de Ejecución:	
Condición(es) para que se ejecute el Caso de Prueba: Se deben cumplir los requerimientos de hardware, software y la configuración de ambiente de pruebas descritos anteriormente.				
Para la Ejecución del Caso de Prueba: Contar con las herramientas citadas anteriormente				
Elemento a probar	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Integración entre IU y modulo lector de registro.	Se debe poder acceder a las funciones del modulo lector de registro desde la IU.	100%	100%	100%
Integración entre IU, modulo lector de registro y modulo generar curva.	Se debe poder generar una(s) curva(s), por medio de la IU, a partir de los datos extraídos del archivo a través del lector.	100%	100%	100%
Integración entre IU, modulo lector de registro y modulo generar imagen	Se debe poder generar una(s) imagen(es), por medio de la IU, a partir de los datos extraídos del archivo a través del lector.	100%	100%	100%
Integración entre IU, modulo lector de registro y modulo generar imagen resistiva	Se debe poder generar una imagen resistiva, por medio de la IU, a partir de los datos extraídos del archivo a través del lector.	100%	100%	100%
Criterios de Aprobación del Caso de Prueba: Se debe cumplir con los resultados esperados en un 100%				
Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo: ___ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba: 31/05/12				

Tabla 2-8. Resultados de pruebas de integración.

Pruebas de sistema

Las pruebas de sistema son realizadas por los programadores del sistema en el ambiente de desarrollo. La Tabla 2-9 describe en qué consisten las pruebas de sistema.

Objetivo:	El objetivo principal de esta prueba es asegurar que el programa realice las funciones especificadas por el cliente en la fase de requerimientos.
Descripción:	En esta prueba se verifica si el sistema cumple con los objetivos planteados (Requisitos funcionales y no funcionales).
Técnicas:	Ejecutar cada CU, cada flujo de CU o función, usando data válida e inválida, para verificar: a) que los resultados ocurran cuando la data sea válida.; b) que se muestren apropiados mensajes de error o alerta cuando data inválida sea empleada.

Fases:	<ol style="list-style-type: none"> 1. Revisión lectura de archivo 2. Revisión graficar curvas 3. Revisión graficar imágenes 4. Revisión graficar imagen resistiva 5. Revisión ver/ocultar grilla 6. Revisión cambiar color 7. Revisión cambiar paleta 8. Revisión hacer zoom
Entorno de prueba:	Se realizará una prueba que verifique el correcto funcionamiento de todas las partes del sistema, usando diferentes archivos de entrada.
Criterios de Éxito:	El sistema funciona adecuadamente si se realizan todas las fases indicadas anteriormente y no ocurre ningún evento inesperado.

Tabla 2-9. Pruebas de sistema.

La Tabla 2-10 muestra en detalle las pruebas de sistema realizadas a VenLog 2.0 con sus respectivos resultados.

ID/Nombre/Sistema/Proyecto: VenLog 2.0		Nivel de Prueba: Parcial		
ID Caso de Uso:		Tipo(s) de Pruebas(s): de sistema		
ID Requerimiento:		Ambiente de Prueba: Nativa		
ID/Nombre Escenario:		Autor del Caso de Prueba: Darwin Martínez		
ID/Nombre Caso de Prueba: CP-02		Nombre del Probador:		
Versión del Caso de Prueba: 1		Fecha de Creación: 23/07/12	Fecha de Ejecución:	
Condición(es) para que se ejecute el Caso de Prueba: Se deben cumplir los requerimientos de hardware, software y la configuración de ambiente de pruebas descritos anteriormente.				
Para la Ejecución del Caso de Prueba: Contar con las herramientas citadas anteriormente				
Funcionalidad a probar	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Leer Archivo	En caso de ser un archivo valido mostrar los datos establecidos en los requerimientos, en caso contrario mostrar un mensaje del error ocurrido (archivo dañado, extensión distinta a .dlis, etc)	100%	100%	100%
Generar curva	Debe permitir construir una curva a través de los datos suministrados.	100%	100%	100%
Ver/Ocultar grilla	Debe permitir mostrar u ocultar una cuadrícula sobre el área de dibujo de la curva.	100%	100%	100%
Cambiar color	Debe permitir cambiar el color de la curva a través de un cuadro de dialogo.	100%	100%	100%
Generar imagen	Debe permitir construir una imagen relativa a la pared del pozo a través de los datos suministrados.	100%	100%	100%

Generar imagen resistiva	Debe permitir combinar los datos de la curva de alineación del norte magnético (azimut) y los datos de los pad de la herramienta resistiva para construir la imagen de las paredes del pozo.	100%	100%	100%
Cambiar paleta	Debe permitir cambiar la tonalidad de la imagen mediante las paletas de colores establecidas.	100%	100%	100%
Hacer zoom	Debe permitir aumentar o reducir el 25% de la escala vertical actual de la imagen o curva. También debe permitir restaurar la escala vertical, la imagen o curva, a su escala original.	100%	100%	100%
Criterios de Aprobación del Caso de Prueba: Se debe cumplir con los resultados esperados en un 100%				
Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo: ____ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba: 11/09/12				

Tabla 2-10. Resultados de pruebas de sistema.

Pruebas de aceptación

Las pruebas de aceptación son realizadas por los clientes o usuarios finales del sistema en dos (2) escenarios distintos, el primer escenario es el entorno de desarrollo donde el usuario final ejecuta el programa en un ambiente controlado por el desarrollador. El segundo escenario es la estación de trabajo de cada uno de los usuarios finales seleccionados para probar el sistema, es un ambiente no controlado por el desarrollador, por lo cual cada usuario deberá documentar los errores e indicárselos al desarrollador posteriormente. La Tabla 2-11 describe en qué consiste las pruebas de aceptación.

Objetivo:	El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad, usabilidad y rendimiento.
Descripción:	Esta prueba es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple los requerimientos establecidos.
Técnicas:	<p><i>Prueba alfa:</i> se lleva a cabo, por un cliente, en el lugar de desarrollo. El cliente usa el software de forma natural mientras el desarrollador lo observa y registra los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.</p> <p><i>Prueba beta:</i> se lleva a cabo por los usuarios finales del software en sus lugares de trabajo. A diferencia de la prueba alfa, el desarrollador no está presente normalmente, por lo tanto la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. Los usuarios registran todos los problemas que encuentren durante la prueba beta e informan a intervalos regulares al desarrollador.</p>

Entorno de prueba:	Esta prueba se realizará con usuarios reales. La aplicación se instalará en los equipos de los usuarios para que ejecuten las pruebas.
Criterios de Éxito:	Si el usuario queda conforme con los resultados obtenidos.

Tabla 2-11. Pruebas de aceptación.

La Tabla 2-12 muestra las pruebas de aceptación realizadas a VenLog 2.0 con sus respectivos resultados.

ID/Nombre/Sistema/Proyecto: VenLog 2.0		Nivel de Prueba: Parcial		
ID Caso de Uso:		Tipo(s) de Pruebas(s): de aceptación		
ID Requerimiento:		Ambiente de Prueba: Estándar		
ID/Nombre Escenario:		Autor del Caso de Prueba: Darwin Martínez		
ID/Nombre Caso de Prueba: CP-03		Nombre del Probador:		
Versión del Caso de Prueba: 1		Fecha de Creación: 23/07/12	Fecha de Ejecución:	
Condición(es) para que se ejecute el Caso de Prueba: Se deben cumplir los requerimientos de hardware, software y la configuración de ambiente de pruebas descritos anteriormente.				
Para la Ejecución del Caso de Prueba: Contar con las herramientas citadas anteriormente				
Aspecto a probar	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Funcionalidad	Se debe cumplir con todos las funcionalidades definidas en la fase de requerimientos.	100%	100%	100%
Usabilidad	Los usuarios llenaran una encuesta denominada <i>System Usability Scale</i> (SUS). (Ver Anexo 1). Según Jeff Sauro, si el promedio del puntaje de las encuestas es mayor a 68% el sistema es usable. (http://www.measuringusability.com/sus.php).	100%	68%	87,5%
Criterios de Aprobación del Caso de Prueba: Los resultados obtenidos deben ser mayor o igual a los resultados esperados.				
Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo: ___ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba: 16/10/12				

Tabla 2-12. Resultados de pruebas de aceptación.

El resultado de la prueba de usabilidad esta avalado por una encuesta, denominada *System Usability Scale* (SUS), la misma se encuentra en el Anexo 4. La encuesta se le realizó a cinco (5) usuarios del sistema y consta con de diez (10) preguntas, cada una tiene cinco opciones de respuesta. Para las preguntas impares se le restará uno a la valoración dada por el usuario y para las preguntas pares la puntuación será de cinco (5) menos la valoración dada por el usuario. De este modo todas las respuestas estarán acotadas entre cero y cuatro. Luego se suman todas las respuestas y se multiplica el valor obtenido por 2.5, para que el rango de las respuestas este entre cero y cien. Adicionalmente hay cuatro (4) preguntas

adicionales que no pertenecen a SUS pero de igual modo intentan medir la usabilidad del sistema.

Durante el desarrollo de esta disciplina se generó el artefacto Plan de Pruebas, el cual incluye el propósito de las pruebas, qué elemento se va a probar y las herramientas a utilizar. En este artefacto también se reflejan las características de hardware y software que serán empleados para realizar el conjunto de las pruebas al sistema. En el Anexo 5 se encuentra el plan de pruebas diseñado para VenLog 2.0. La Figura 2-7 ilustra cómo se integraron las distintas herramientas de software para la ejecución del conjunto de pruebas realizadas a la aplicación.

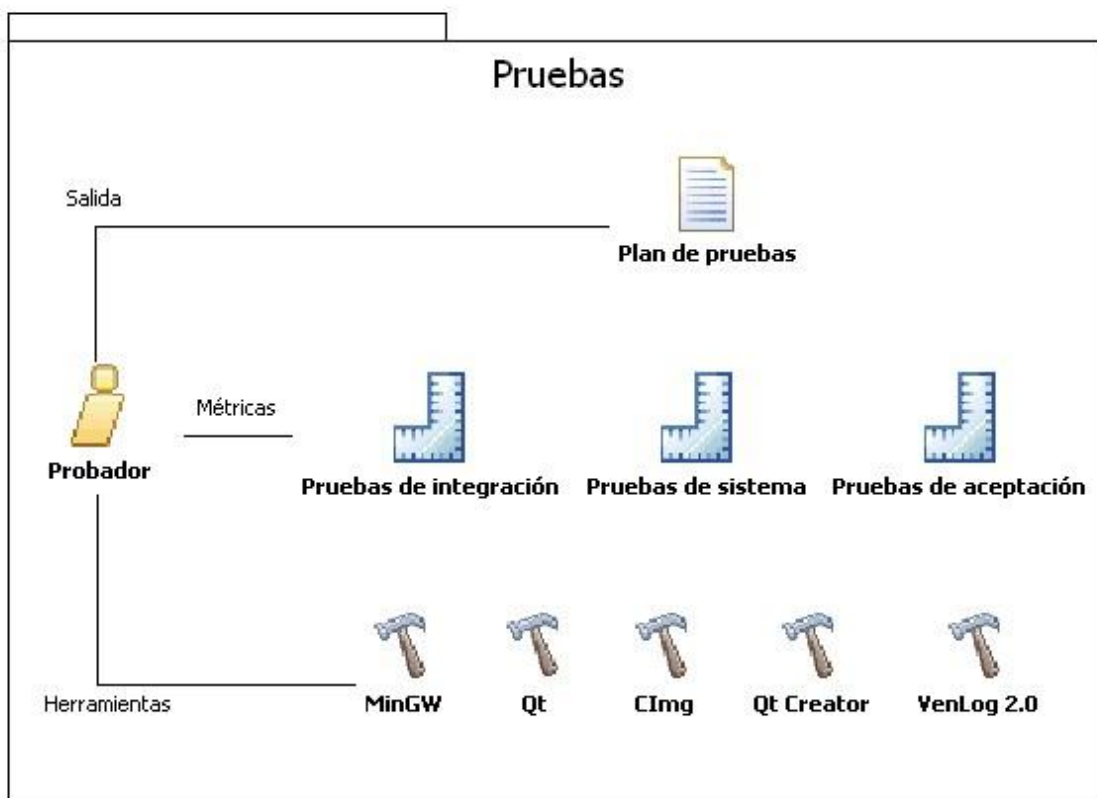


Figura 2-7. Integración de la metodología con las herramientas de software en la disciplina pruebas.

En la Figura 2-7 ilustra como el probador haciendo uso de las herramientas de software puede realizar las diversas pruebas de la aplicación, utilizando métricas diferentes para cada tipo de prueba (unitarias, de integración, de sistema o de aceptación) para generar el artefacto plan de pruebas correspondiente a esta disciplina.

En este capítulo se definió el modelo de procesos MeRinde, también se describió la disciplina requerimientos, en la cual se definieron tanto los requerimientos funcionales como los no funcionales de la aplicación VenLog 2.0, se definió el diseño de la arquitectura y se presentó el diagrama de clases del módulo de lectura, se describieron las herramientas de software utilizadas para la implementación de VenLog 2.0 y por último se detallaron las pruebas que se le realizaron a la aplicación.

En el siguiente capítulo se muestra varios recorridos por VenLog 2.0 mediante la descripción de tres (3) escenarios de uso.

Capítulo 3

Escenarios de uso de la herramienta VenLog 2.0

Un escenario es una descripción parcial y concreta del comportamiento de un sistema en una determinada situación. Es una descripción parcial, porque no necesita describir todas las características de las entidades involucradas, sólo se describe aquello que está relacionado con un comportamiento particular del sistema analizado. A pesar de estar acotados a un determinado comportamiento, describen todo el contexto que involucra a esa actividad: recursos del sistema, objetivos de los usuarios, contexto social en que se desarrolla y entidades involucradas. Proveen un “retrato” de como esa actividad se lleva a cabo. Los escenarios describen situaciones teniendo en cuenta aspectos de uso, permitiendo: conocer el problema, unificar criterios, ganar compromiso con clientes/usuarios, organizar los detalles involucrados y entrenar a nuevos participantes (UNICEN, 2012).

En este capítulo se muestra varios recorridos por VenLog 2.0 mediante la descripción de tres (3) escenarios de uso y las posibles combinaciones entre éstos, además de describir el asistente de instalación y desinstalación de ésta herramienta de software.

3.1 Instalación y desinstalación de VenLog 2.0

La herramienta VenLog 2.0 fue empaquetada en un archivo ejecutable, la cual contiene los archivos necesarios para que la aplicación funcione fuera del ambiente de desarrollo. Éste archivo ejecutable permite la instalación de VenLog 2.0, colocando un acceso directo en el menú de inicio y en el escritorio.

El proceso de instalación de inicia al hacer doble clic sobre el archivo ejecutable que tiene la siguiente apariencia:



La Figura 3-1 muestra la ventana de bienvenida del asistente de instalación de VenLog 2.0 para continuar con la instalación de debe presionar el botón siguiente.



Figura 3-1. Ventana de bienvenida del asistente de instalación de VenLog 2.0.

Al presionar el botón siguiente se muestra otra ventana (ver Figura 3-2) en la cual se debe indicar la ruta de instalación, de no ser así se instalará en la ruta establecida por defecto (C:\Archivos de programa\PDVSA\Intevp).



Figura 3-2. Ruta de instalación.

Al presionar instalar se empezarán a extraer los archivos empaquetados en el ejecutable en la carpeta destino, la Figura 3-3 muestra un ejemplo de esto.



Figura 3-3. Extracción de archivos.

Al terminar de extraer todos los archivos se activa el botón de siguiente, al presionarlo aparecerá la ventana que indica que se ha completado la instalación (ver Figura 3-4) se presiona el botón de terminar y la aplicación estará disponible en el equipo, se puede acceder a ésta por medio del menú de inicio o por medio de un acceso directo que se crea en el escritorio durante la instalación.



Figura 3-4. Finalizando la instalación.

Así como VenLog 2.0 cuenta con un asistente para su instalación, del mismo modo para su desinstalación también se cuenta con un asistente. Éste asistente puede ser accedido desde el menú de inicio en la ruta /VenLog 2.0/Desinstalar. La Figura 3-5 muestra la ventana de bienvenida del asistente para la desinstalación de VenLog 2.0



Figura 3-5. Asistente de desinstalación.

Al presionar el botón siguiente se muestra otra ventana (ver Figura 3-6) la cual muestra la ruta donde se encuentra instalada la aplicación (D:\Archivos de programa\PDVSA\Intevp) y da tres (3) opciones, devolverse a la pantalla de bienvenida, desinstalar o cancelar.




Figura 3- 6. Desinstalación de VenLog 2.0.

Al presionar desinstalar se borrarán los archivos del equipo, al finalizar la desinstalación se muestra la pantalla de finalización del asistente (ver Figura 3-7)



Figura 3-7. Finalizando la desinstalación.

Para ejecutar cada uno de los escenarios de uso es necesario realizar unos pasos previos los cuales se explican a continuación.

Se debe acceder a VenLog 2.0 a través del icono  que se genera automáticamente durante el proceso de instalación tanto en el menú de inicio como en el escritorio.

Cuando se inicia VenLog 2.0 se visualiza el marco de trabajo principal como se puede apreciar en la Figura 3-8

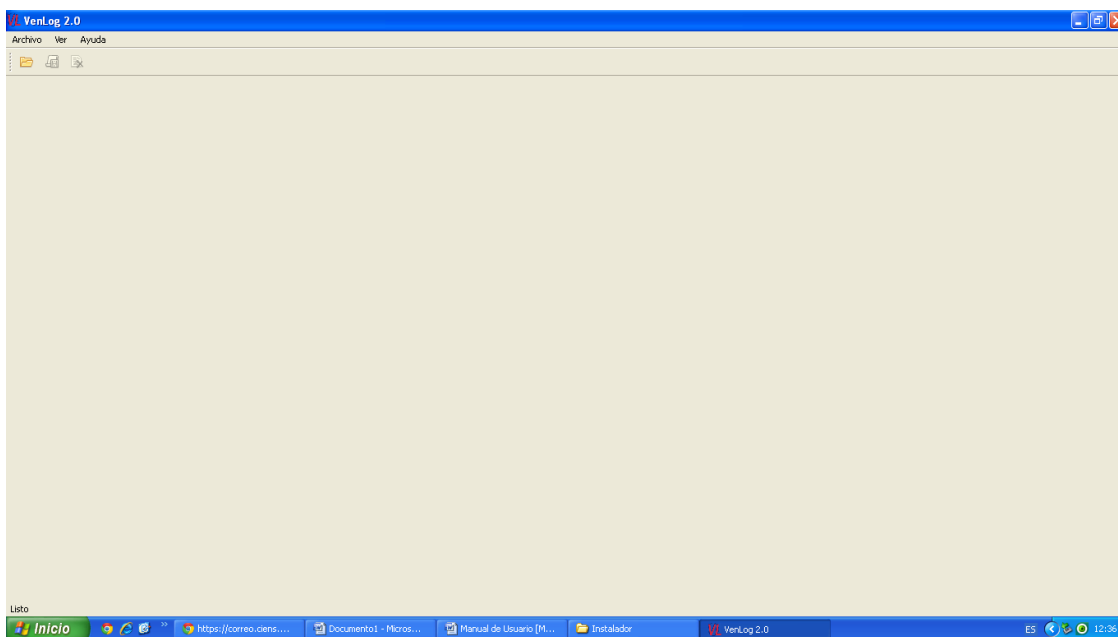



Figura 3- 8. Marco de trabajo de VenLog 2.0.


El marco de trabajo principal pone a disposición los siguientes elementos:


1. El menú principal de la aplicación, el cual está ubicado en la esquina superior izquierda y se organiza de la siguiente manera:


- *Archivo*: ofrece las siguientes opciones:



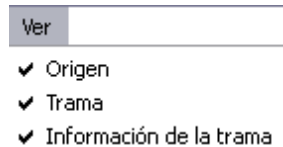
 *Abrir archivo*: Permite abrir un archivo de tipo DLIS. Si existe un archivo abierto, se cerrará el archivo actual y se abrirá el nuevo archivo.

 *Cerrar archivo*: Permite cerrar el archivo sin cerrar la instancia del programa.

 *Exportar registro*: Permite exportar los datos del archivo (origen, tramas y canales) en formato HTML. (esta opción sólo se activa si hay un archivo abierto)

 *Salir*: Permite salir de la aplicación.

- *Ver*: permite ocultar y/o mostrar los distintos paneles de la aplicación.



- *Ayuda*: muestra la ayuda de VenLog 2.0.




NOTA: El menú *Ver* y las opciones *exportar* y *cerrar* del menú *Archivo* sólo se habilitan cuando se encuentra un archivo abierto.

2. La barra de herramientas contiene un grupo de iconos que permiten el acceso directo a funciones del menú *Archivo*. Estos iconos se activan o desactivan dependiendo de la instancia del programa donde se encuentre según el flujo de trabajo.



3. Una barra de estado, donde se visualizan los mensajes generados por VenLog 2.0. La misma se ubica en la parte inferior izquierda del marco de trabajo principal. Esta modalidad de visualización de los mensajes viene a sustituir a los tradicionales mensajes emergentes, los cuales resultan incómodos y poco elegantes.

Listo

Para continuar con la ejecución del programa se debe abrir un archivo de DLIS, esto se hace por medio de la opción *Abrir archivo* presente en el menú *Archivo* o por medio del siguiente icono  presente en la barra de herramientas. Esta opción activa una ventana

emergente en la cual el usuario podrá indicar la ruta del archivo como se muestra en la Figura 3-9.

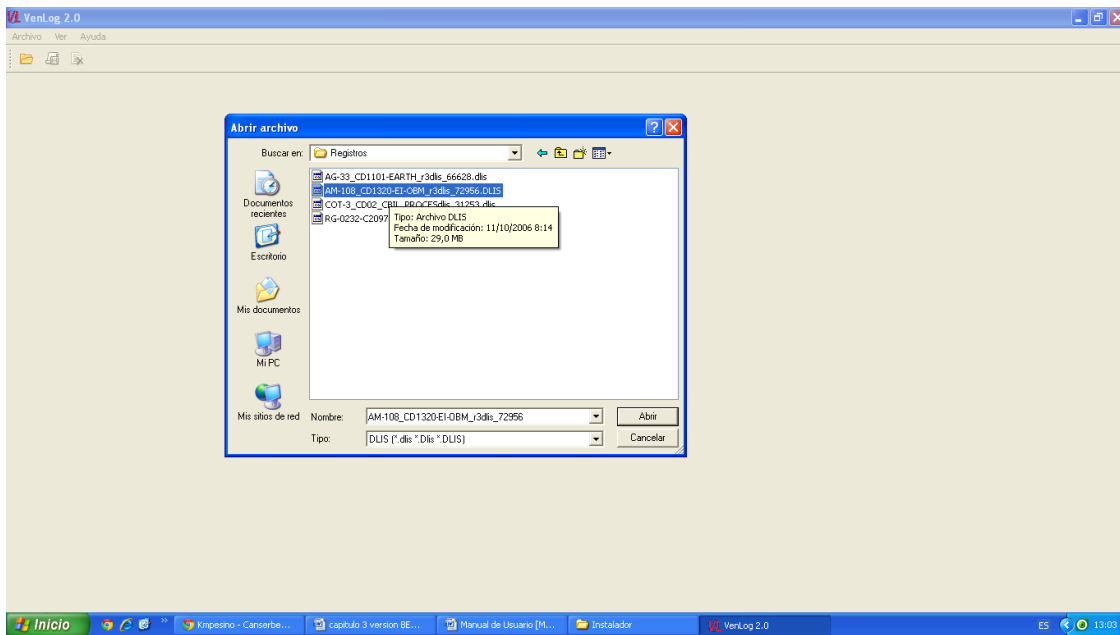


Figura 3-9. Ventana emergente que permite indicar la ruta del archivo a procesar.

Al abrir el archivo se mostraran dos (2) paneles: El panel Trama, que indica el nombre de las tramas presentes en el archivo, como se muestra en la Figura 3-10

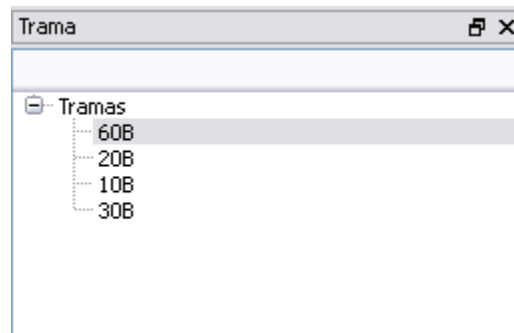


Figura 3- 10. Panel Trama.

El panel Origen, que contiene la información relevante sobre los datos de origen de la muestra (compañía de servicio que realizó el perfilaje, software con que se tomo grabó el perfil, versión del software, pozo en el que fue tomada la muestra, entre otros) como se puede apreciar en la Figura 3-11.

Origen	
Identificador de archivo	USI_SONIC_038PUP
Conjunto de nombre	PDVSA/RG-232
Producto	OP
Versión	10C0-306
Herramienta	WELLCAD: WellCAD file generator
Compañía de servicio	PDVSA
Tipo de registro	SLB

Figura 3-11. Panel Origen.

Para realizar las tareas de graficación de imágenes y/o curvas, es necesario que se seleccione una trama haciendo doble clic sobre alguno de los ítems presentes en el panel trama, esto habilita el panel que contiene información de la trama como: nombre de la trama, cantidad de canales que posee, inicio, fin y dirección en la que fue tomado el perfil (ascendente o descendente). Y una vista de datos, perteneciente al área central, la cual contiene una tabla con las propiedades de los canales contenidos en la trama, dos casillas de verificación por cada canal, y una barra de herramienta, la cual contiene iconos que permiten graficar los canales seleccionados como se muestra en la Figura 3-12.

The screenshot shows the VenLog 2.0 software interface. The 'Origen' panel on the right contains the following data:

Origen	
Identificador de archivo	USI_SONIC_038PUP
Conjunto de nombre	PDVSA/RG-232
Producto	OP
Versión	10C0-306
Herramienta	WELLCAD: WellCAD file generator
Compañía de servicio	PDVSA
Tipo de registro	SLB

The central table lists channel properties:

Generales	Resistivos	Nombre	Unidad	Dimensión	Código	Característica
<input type="checkbox"/>	<input type="checkbox"/>	TDEP	0.1 in	1	2	6-Inch Frame Depth
<input type="checkbox"/>	<input type="checkbox"/>	BS	in	1	2	Bit Size
<input type="checkbox"/>	<input type="checkbox"/>	CS	R/h	1	2	Cable Speed
<input type="checkbox"/>	<input type="checkbox"/>	CVEL	R/min	1	2	Cable Velocity
<input type="checkbox"/>	<input type="checkbox"/>	TENS	lbf	1	2	Cable Tension
<input type="checkbox"/>	<input type="checkbox"/>	ETIM	s	1	2	Elapsed Logging Time
<input type="checkbox"/>	<input type="checkbox"/>	CUCHV	V	1	2	Uncalibrated Head Voltage
<input type="checkbox"/>	<input type="checkbox"/>	CFVL	us/ft	1	2	Fluid velocity DFFM
<input type="checkbox"/>	<input type="checkbox"/>	CZMD	1000 kPa.s/m	1	2	Acoustic Impedance of Mud (DFFM)
<input type="checkbox"/>	<input type="checkbox"/>	UCAZ	deg	1	2	USIT Azimuth
<input type="checkbox"/>	<input type="checkbox"/>	RB_USIT	deg	1	2	Memorized Relative Bearing for USI
<input type="checkbox"/>	<input type="checkbox"/>	UFRT	Hz	1	2	Firing rate
<input type="checkbox"/>	<input type="checkbox"/>	UTDL	us	1	2	Tachymetry delay
<input type="checkbox"/>	<input type="checkbox"/>	DCHV	V	1	2	Transmitter DC high voltage
<input type="checkbox"/>	<input type="checkbox"/>	ACLV	V	1	2	Transmitter AC low voltage
<input type="checkbox"/>	<input type="checkbox"/>	ACHV	V	1	2	Transmitter AC high voltage
<input type="checkbox"/>	<input type="checkbox"/>	RSRV	c/s	1	2	Motor revolution speed
<input type="checkbox"/>	<input type="checkbox"/>	MOTV	V	1	2	Motor voltage

The 'Información de la trama' panel on the right shows the following data:


Información de la trama	
Nombre de la trama	60B
Origen	35
Copia	0
Número de canales	78
Inicio	94807.1 ft
Fin	79613.1 ft
Dirección	DECREASING

Figura 3-12. Entorno de VenLog 2.0 con un archivo abierto.

Hasta aquí son los pasos comunes para ejecutar los diferentes escenarios de uso, a continuación se explica cada uno de éstos.

3.2 Escenario de uso 1: Graficar curva(s)

El escenario Graficar curva tiene como objetivo indicar los pasos necesarios para graficar una o más curvas y realizar las operaciones disponibles, tales como: ver u ocultar cuadrícula, cambiar color, aumentar o reducir escala vertical y superponer curvas.

Para graficar una o más curvas se deben seleccionar los canales correspondientes (deben ser de dimensión uno) en la casilla de verificación de la primera columna de la tabla y presionar el botón de graficar curvas e imágenes (), una vez presionado el botón se mostrarán las curvas asociadas a los canales seleccionados como se puede apreciar en la Figura 3-13.

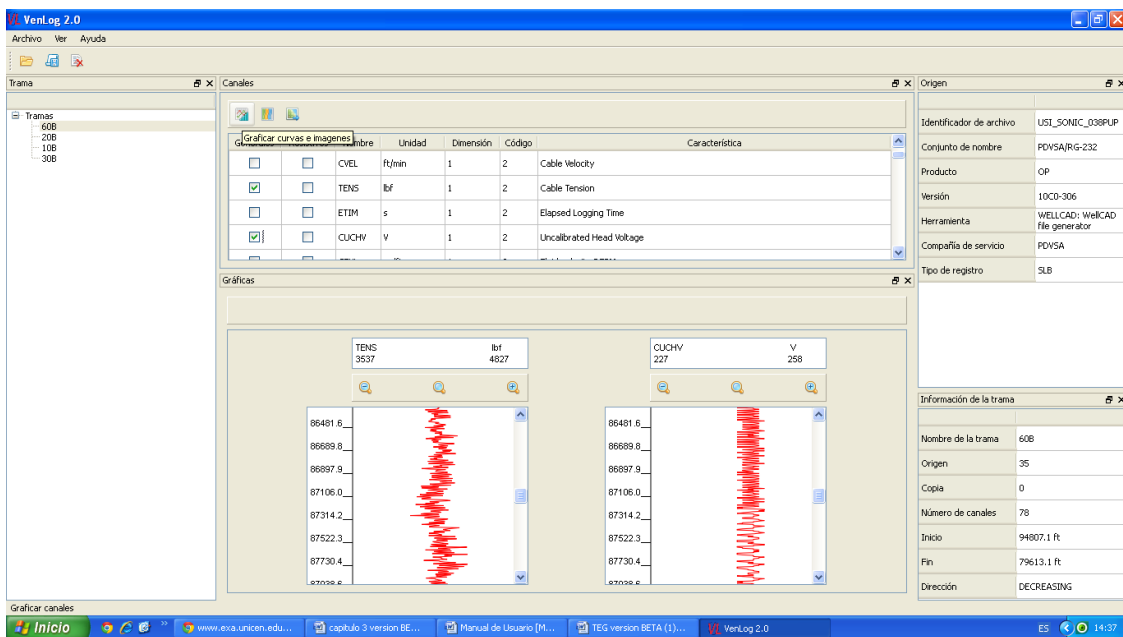


Figura 3-13. Selección de canales (Curva).

Al hacer clic derecho sobre el área de dibujo de la curva se despliega un menú con las siguientes opciones: ver/ocultar grilla y cambiar color, como se puede observar en la Figura 3-14. La primera, muestra u oculta una cuadrícula sobre el área de dibujo de la curva; mientras que la segunda activa una ventana emergente que permite cambiar el color de la curva.

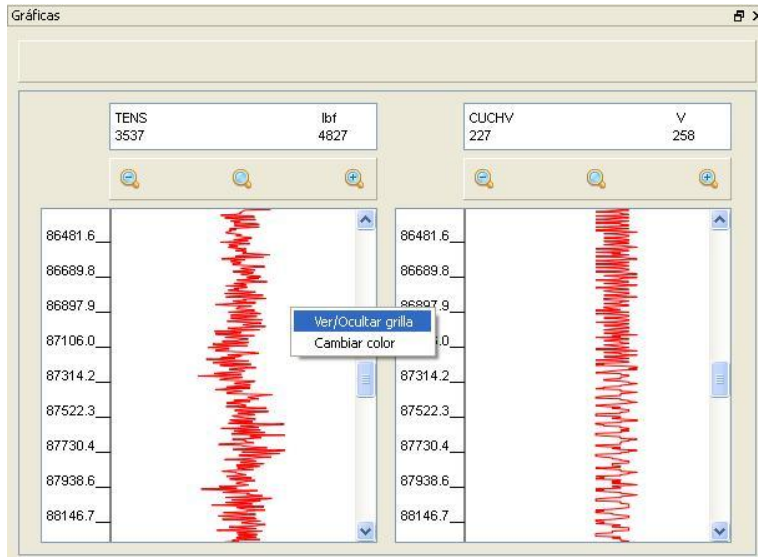


Figura 3-14. Menú de las curvas.

En la Figura 3-15, la curva del lado izquierdo tiene visible la cuadrícula, mientras que la curva de la derecha tiene activa la ventana emergente para cambiar el color de ésta.

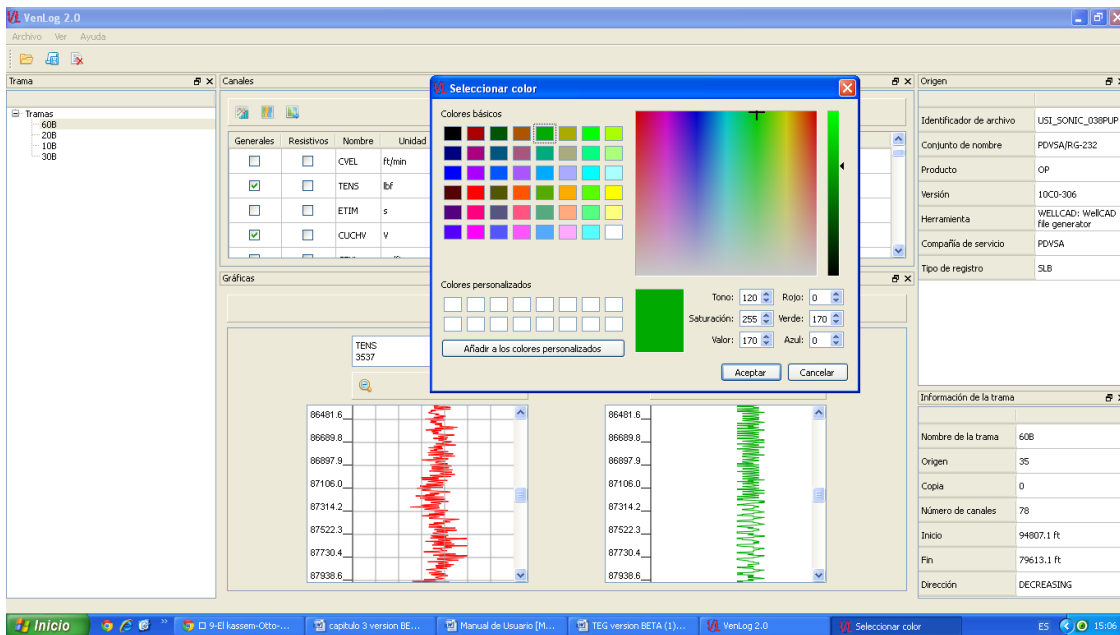





Figura 3- 15. Cambiar mostrar cuadrícula y cambiar color de la curva.

Para cambiar la escala vertical de alguna de las curvas se utilizan los iconos en forma de lupa que están sobre ésta (reducir tamaño , restaurar tamaño  y aumentar tamaño ). En la Figura 3-16, la escala vertical de la curva de la izquierda se redujo cuatro (4) veces, mientras que a la curva de la derecha se aumentó cuatro (4) veces su escala vertical. Como

se puede observar las opciones de cambio de escala permiten contraer o expandir los trazos de la curva sin modificar su resolución.

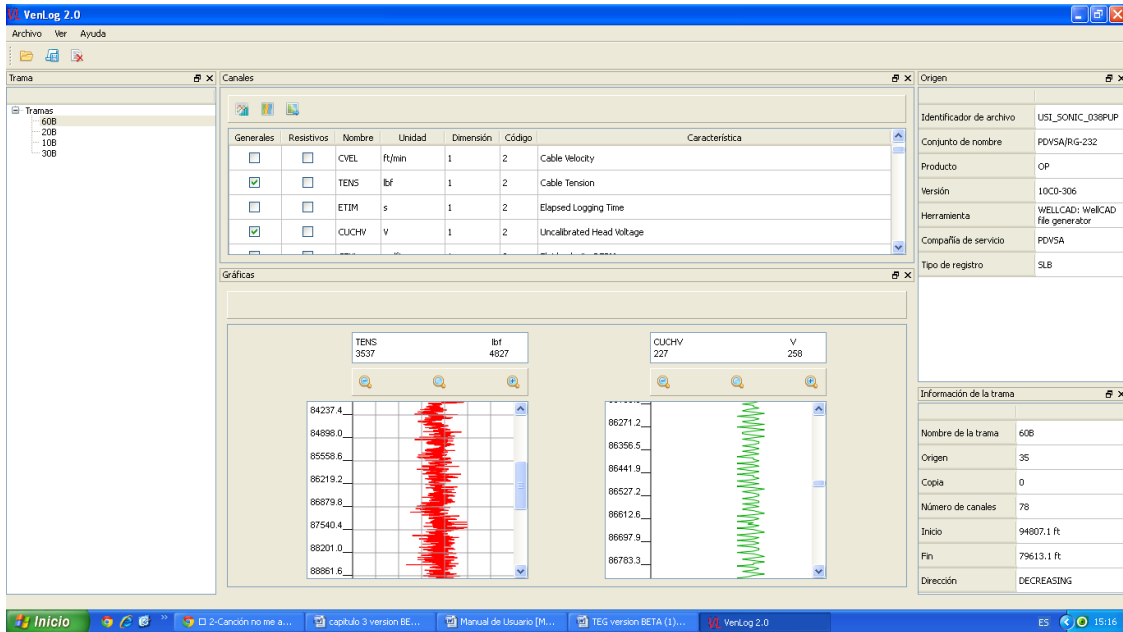



Figura 3-16. Cambiar escala vertical.

En ocasiones hay curvas muy similares y es necesario establecer las diferencias entre éstas, por lo cual se dispone de la opción superponer curvas, la cual se ejecuta al presionar el botón de graficar curvas superpuestas (), esto dibujará las curvas seleccionadas en una sola área de dibujo. En la Figura 3-17 del lado izquierdo se encuentra el área de dibujo con las curvas, mientras que del lado derecho se encuentra un rectángulo por cada curva mostrada, en el cual se indica el nombre de la curva, la escala, el color y la unidad. Para cambiar de color alguna de las curvas se hace clic derecho sobre alguno de los rectángulos de la derecha y se selecciona cambiar color. Para ver u ocultar la cuadrícula se hace clic derecho sobre el área de dibujo de las curvas.

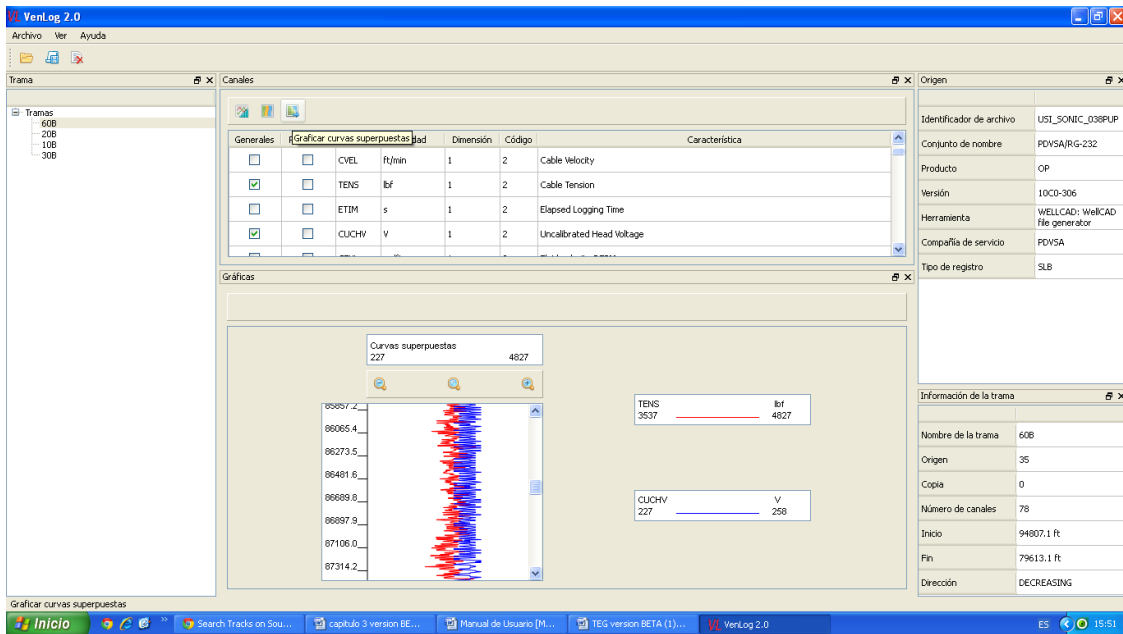



Figura 3- 17. Curvas superpuestas.

3.3 Escenario de uso 2: Graficar imagen

El escenario Graficar imagen tiene como objetivo indicar los pasos necesarios para graficar una o más imágenes de tipo sónica y realizar las operaciones disponibles, tales como: cambiar paleta de colores y aumentar o reducir escala vertical.

Para graficar una o más imágenes se deben seleccionar los canales correspondientes (deben ser de dimensión mayor a uno) en la casilla de verificación de la primera columna de la tabla y presionar el botón de graficar curvas e imágenes (), una vez presionado el botón se mostrarán las imágenes asociadas a los canales seleccionados como se puede apreciar en la Figura 3-18.

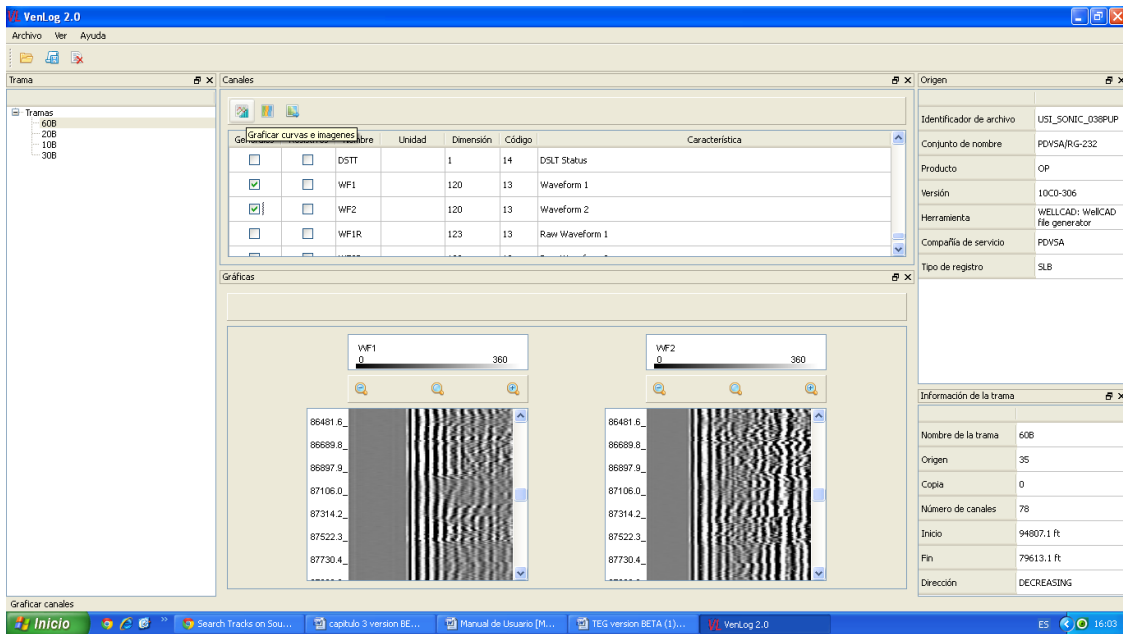


Figura 3-18. Selección de canales (Imagen)

Dependiendo del rango de valores con los que se construya una imagen se le asigna una paleta de colores por defecto. En ocasiones es necesario cambiar esta paleta para determinar o visualizar de manera más clara las posibles fallas geológicas presentes en la estructura. La paleta de colores puede ser cambiada al hacer clic sobre el rectángulo blanco que se encuentra sobre la imagen. En la Figura 3-19 se pueden visualizar dos (2) de las paletas disponibles.

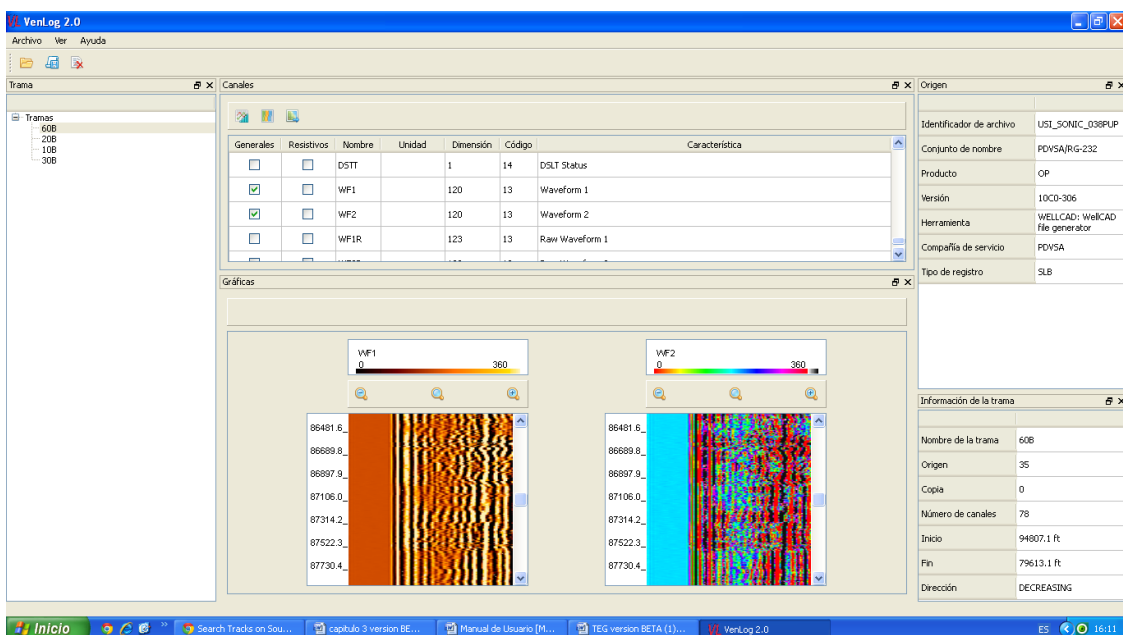






Figura 3- 19. Cambiar paleta de colores de la imagen.

Otra de las operaciones que se puede hacer con las imágenes es la de cambiar la escala vertical. Esto se hace de la misma manera que en el escenario anterior, presionando los iconos en forma de lupa que están sobre la imagen (reducir tamaño , restaurar tamaño  y aumentar tamaño ).

3.4 Escenario de uso 3: Graficar imagen resistiva

El escenario Graficar imagen resistiva tiene como objetivo indicar los pasos necesarios para graficar una imagen de tipo resistiva.

Para graficar una imagen resistiva se deben seleccionar, en la casilla de verificación de la segunda columna de la tabla, los cuatro (4), seis (6) u ocho (8) canales que correspondan a los patines de la herramienta de perfil de imagen resistiva y el canal que indica el azimut de orientación del primer patín. Al presionar el siguiente botón  se mostrará la imagen resistiva asociada a los canales seleccionados como se puede apreciar en la Figura 3-20.

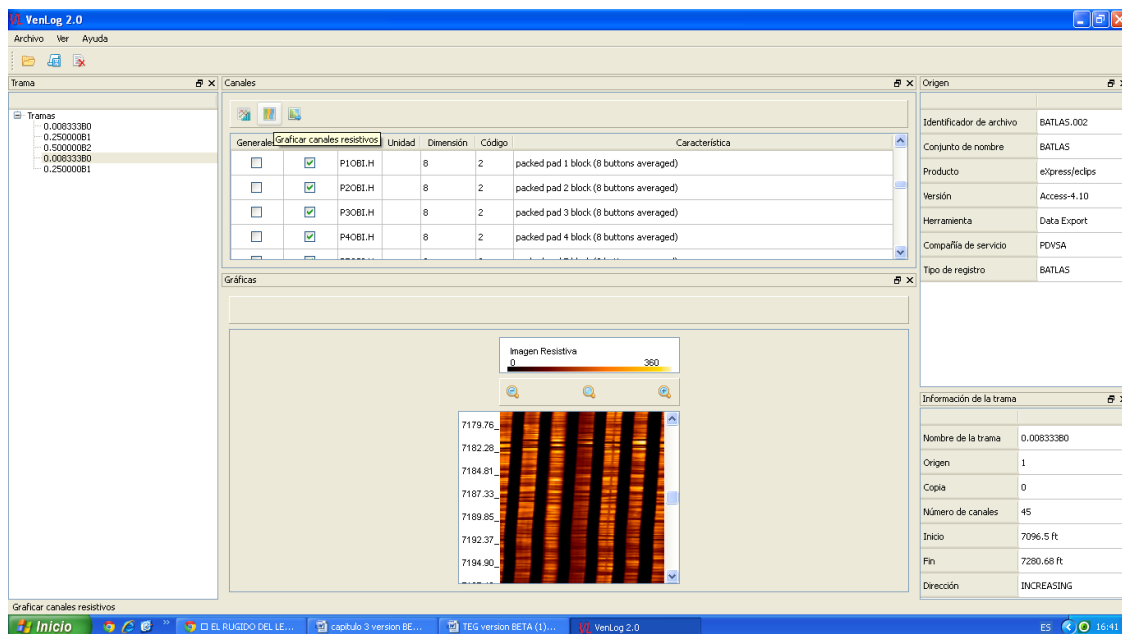



Figura 3-20. Selección de canales (Imagen resistiva).

Al igual que en el escenario anterior esta imagen cuenta con una paleta de colores que puede ser cambiada a gusto del usuario, también cuenta con los iconos en forma de lupa que permiten cambiar la escala vertical de la imagen.

Por lo general la visualización de los perfiles de imagen, ya sea sónico o resistivo, contienen un dato adicional que es la curva de *Gamma ray*, con la cual se puede determinar

el tipo de arcilla y detectar los minerales pesados. En vista de esto y pensando en que el usuario pueda necesitar aparte del *Gamma ray*, visualizar otra curva junto con las imágenes o hacer comparaciones entre imágenes resistivas e imágenes sínicas, se pueden combinar los tres (3) escenarios presentados anteriormente.

Si se desea visualizar curvas e imágenes sínicas, se deben seleccionar los canales correspondientes en la casilla de verificación de la primera columna de la tabla y presionar el botón graficar curvas e imágenes (), luego se mostrará las curvas e imágenes asociadas a los canales seleccionados como se puede apreciar en la Figura 3-21.

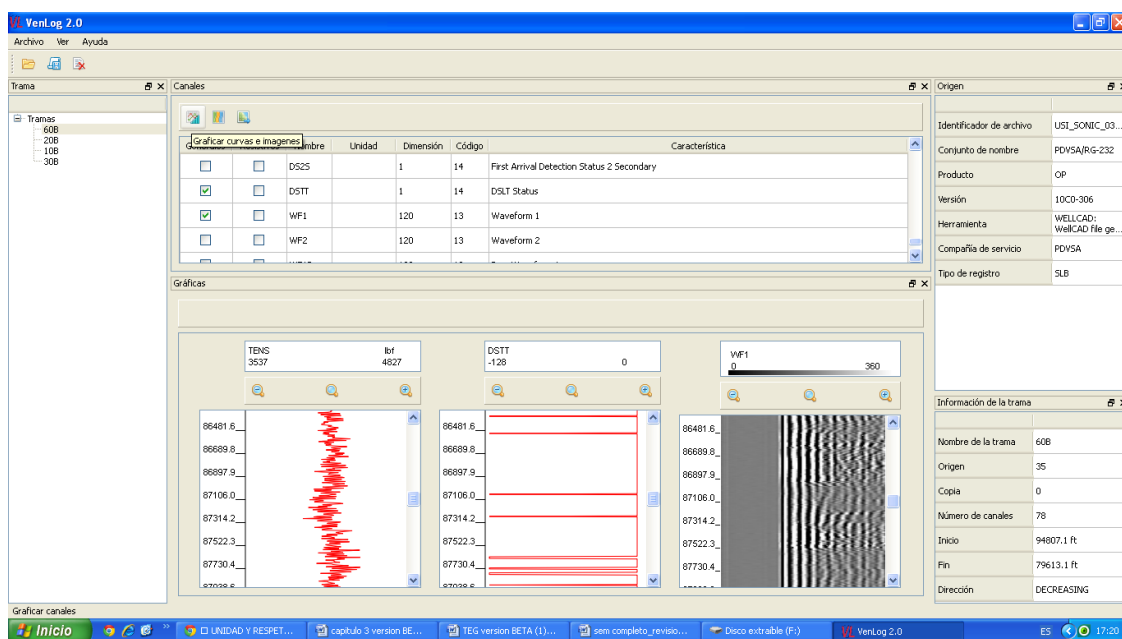



Figura 3-21. Selección de canales (curvas e imágenes).

Si se desea visualizar curvas, imágenes sínicas e imagen resistiva, se deben seleccionar los canales correspondientes a las curvas e imágenes sínicas en la casilla de verificación de la primera columna de la tabla y los canales correspondientes a la imagen resistiva se deben seleccionar en la casilla de verificación de la segunda columna de la tabla, y presionar el botón graficar canales resistivos (), luego se mostrarán las curvas, imágenes sínicas e imagen resistiva asociada a los canales seleccionados como se puede apreciar en la Figura 3-22.

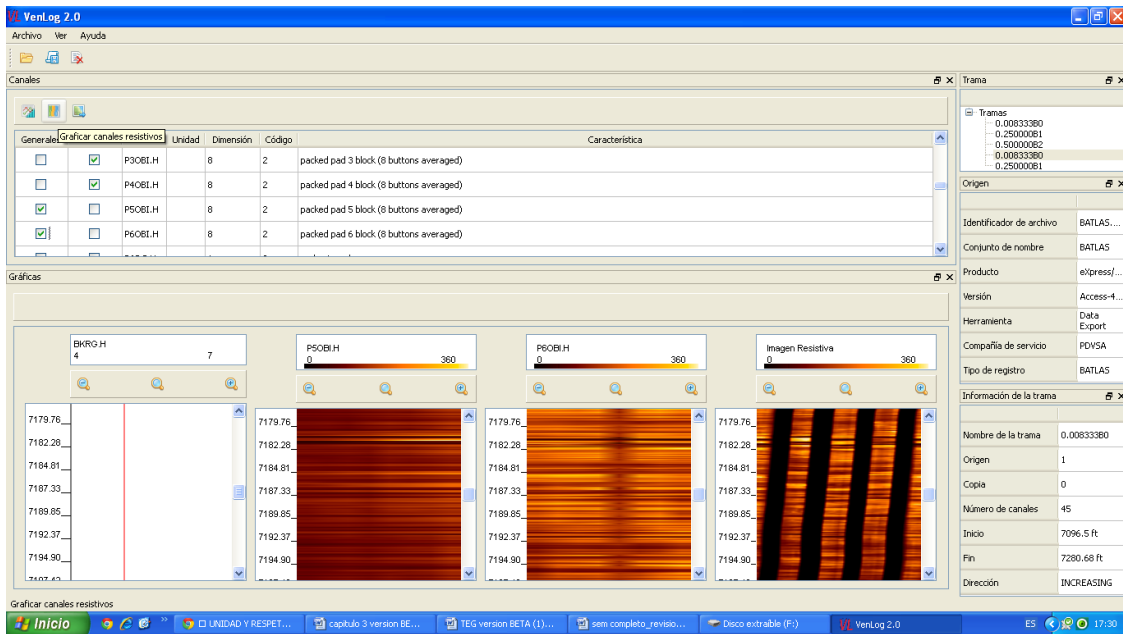


Figura 3- 22. Selección de canales (curvas, imágenes e imagen resistiva).

En este capítulo se describió el entorno de la herramienta VenLog 2.0 y se plantearon los tres (3) escenarios de uso básicos, los cuales se ejecutaron bajo el Sistema Operativo Windows Xp en una máquina con un procesador AMD Sempron (tm) 130 2,61GHz y 896 MB de memoria RAM.

A continuación de presentarán las conclusiones de este Trabajo Especial de Grado.

Conclusiones y recomendaciones

Con el propósito de apoyar el estudio de estabilidad de hoyos realizado por parte de la Gerencia Técnica de Producción y Construcción de Pozos (PRCP) de PDVSA Intevep en este Trabajo Especial de Grado se desarrolló el módulo Lector de Registros DLIS para la aplicación VenLog en su versión 2.0. Esta herramienta permite la lectura, interpretación y procesamiento de perfiles sísmicos, resistivos y de imagen almacenados en archivos del tipo DLIS versión uno (1). De este modo se da cumplimiento al objetivo general de este trabajo.

Para el desarrollo del software se utilizó MeRinde como modelo de proceso, el mismo permitió guiar las actividades y el desarrollo de tareas para cada una de sus disciplinas, quedando todo debidamente documentado en los respectivos artefactos por disciplina. Estos últimos se pueden ver con mayor detalle en los Anexos 1, 2 y 5.

El principal problema presentado durante el desarrollo de este Trabajo Especial de Grado fue entender el comportamiento del estándar DLIS, especialmente la relación que existen entre un registro lógico con formato explícito (EFLR) y un registro lógico con formato indirecto (IFLR). Ambos registros están vinculados por un identificador único. El IFLR contiene un conjunto de bytes sin formato, en el cual está contenida la información relevante del perfilaje, con estos datos es que se puede construir las gráficas (imágenes y curvas). El EFLR nos da el formato en el que se deben traducir el conjunto de bytes del IFLR.

Otro problema presentado fue el tiempo invertido en la familiarización con el Framework Qt, ya que no se tenía experiencia previa al momento de empezar el proyecto y esto implicaba superar la curva de aprendizaje.

El Módulo Lector de Registros de VenLog 2.0, permite generar imágenes y/o curvas correspondientes a los datos seleccionados por el usuario. Entre las operaciones con las imágenes y/o curvas están: cambiar el color y aumentar o disminuir la escala vertical para observar con un mayor nivel de detalle la imagen o curva. Es justo decir que, en el alcance del proyecto no estuvo previsto que la aplicación realizara cálculos asociados a fallas y fracturas que estén presente en la estructura del pozo; sin embargo, estos cálculos estarán entre los requerimientos para desarrollos a futuro, teniendo como base el Módulo Lector de Registros desarrollado para este Trabajo Especial de Grado.

VenLog 2.0 ajusta y unifica la metodología del procesamiento de los datos dentro del programa, independientemente de la fuente propietaria: Baker, Schlumberger, Halliburton⁷, o cualquier otra compañía de servicio, con la cual se garantiza la independencia en la interpretación de archivos DLIS.

De igual modo, a pesar de que este trabajo está enfocado en la interpretación de registros de imágenes sísmicos y resistivos, esta herramienta puede interpretar cualquier archivo de tipo DLIS.

Con la culminación y puesta en producción de este proyecto se logra apoyar a la optimización del análisis e interpretación de estos registros, ya que elimina la dependencia de terceros y amplía e impulsa la ejecución de nuevos proyectos tanto en el área de investigación y desarrollo como asistencia técnica especializada.

Cabe destacar que el grupo de desarrollo a futuro debería estudiar la factibilidad de extender el lector para interpretar archivos DLIS con la versión dos (2) del estándar y archivos de tipo LAS, además estudiar la posibilidad de presentar las gráficas en un formato 3D, ya que los usuarios se mostraron interesados en este tipo de visualización y exportar los datos con los que se construyen las gráficas (imágenes y/o curvas) a formato txt, para realizar análisis de distribución de datos.

⁷ Baker, Schlumberger, Halliburton son compañías de servicios contratadas por PDVSA para la toma y monitoreo de perfiles de pozos.

Referencias bibliográficas

- Amaya. (2012). *cplusplus.com*. Obtenido de <http://www.cplusplus.com/info/description/>
- Baker Atlas. (2006). *Workshop de registros acusticos e imágenes*. PDVSA.
- Baker Hughes. (2006). *Applications of borehole image technology*. Baker Atlas.
- Baker Hughes. (2008). *Borehole Imaging Services and Applications*. Baker Atlas.
- Chumbimuni, R. (2008). *Desarrollo del Lector de Archivo DLIS*. Los Teques.
- Corporation, N. (2011). *qt.nokia.com*. Obtenido de <http://doc.qt.nokia.com/qtcreator-2.4/creator-overview.html>
- Diaz, E. (2006). *Introducción de los registros de imágenes: Control de calidad y procesamiento*. PDVSA.
- Gabardini, J. (2004). *Balanceo de Metodologías Orientadas al Plan y Ágiles Herramientas para la Selección y Adaptación*. Buenos Aires, Argentina: PMI Global Congress Proceedings.
- Gomez, O. (1975). *Registros de pozos: Teoría e interpretación Parte 1*. México DF: Primera edición.
- Guyod, H. (1974). *Perfil Eléctrico: Introducción al perfilaje*. USA: DRESSER ATLAS.
- Marrero, C., & Santos, K. (2007). *METODOLOGÍA DE LA RED NACIONAL DE INTEGRACIÓN Y DESARROLLO DE SOFTWARE LIBRE (Guía Detallada)*. CNTI.
- MinGW.org. (Mayo de 2012). *Home of the MinGW and MSYS Projects*. Obtenido de http://www.mingw.org/Welcome_to_MinGW_org
- Martínez, D. (2012). *Informe de pasantías - Revisión de aplicación para la lectura y visualización de registros acústicos y resistivos (VenLog)*. Caracas.
- Montilla, G. (2006). *Plugin Lector DLIS VI*. Documentación técnica de la gerencia de AIT - Intevep.

OMG. (Enero de 2012). MOF - Meta Object Facility. Obtenido de <http://www.omg.org/mof/>

OMG. (Mayo de 2012). SPEM - Software Process Engineering Metamodel. Obtenido de <http://www.omg.org/spem/>

PDVSA. (2005). *Sitio oficial de Petróleos de Venezuela S.A.* Obtenido de <http://www.pdvsa.com>

POSC. (1998). *energistics.org*. Recuperado el 15 de noviembre de 2012, de <http://w3.energistics.org/rp66/v1/Toc/main.html>

Salazar, C. (2011). *Especificación del modelo de proceso MeRinde haciendo uso de la notación SPEM*. Caracas: Seminario de postgrado de Ciencias de la Computación UCV.

Schlumberger Limited. (1970). Fundamentos de la interpretación de perfiles.

Schlumberger Limited. (1972). Log Interpretation Volume I - Principles. New York.

Smolen, J. (1995). Cased - Hole and Production Log Evaluation. Pennwell Pub.

StarUML. (Enero de 2008). *sourceforge.net*. Obtenido de <http://staruml.sourceforge.net/en/about.php>

Tittman, J. (1986). Geophysical Well Logging: Excerpted from Method of Experimental Physics, Volume 24, Geophysics. Academic Press Inc.

Tschumperlé, D. (Octubre de 2004). *CImg*. Obtenido de <http://cimg.sourceforge.net/index.shtml>

UNICEN. (2012). *unicen*. Recuperado el Septiembre de 2012, de http://www.exa.unicen.edu.ar/catedras/ingrequi/index_archivos/Notas-Escenarios.pdf

Wichmann, P. (1974). Perfil Acústico: Introducción al Perfilaje. USA: DRESSER ATLAS.

Zambrano, C. (Noviembre de 2011). Funciones de las gerencias PRCP y AIT. (D. Martínez, Entrevistador)

Anexo 1



Gerencia AIT
(Automatización, Informática y
Telecomunicaciones)

Especificación de Requerimientos del
Software

< VenLog 2.0 >

Versión: < 1.1.0 >

< Febrero, 2012 >

Descripción del Documento

El objetivo del ERS es documentar todos los requerimientos del sistema, este describe las funciones del sistema, los requerimientos no funcionales, características del diseño, y otros elementos necesarios para proporcionar una descripción completa y comprensiva de los requerimientos para el software a desarrollar.

Los requerimientos pueden ser levantados con diferentes herramientas, también se pueden encontrar dispersos en varios artefactos y herramientas. Es por ello, que esta metodología propone capturar todos los requerimientos para el ERS en un solo artefacto, el cual está conformado por dos (2) artefactos que describen los requerimientos que son: Modelo de Casos de Uso y Especificaciones Suplementarias.

El artefacto ERS controla la evolución del sistema durante todo el ciclo de desarrollo del proyecto, cuando las nuevas características son añadidas o modificadas al artefacto de visión, son aclarados dentro del artefacto ERS.

Las decisiones hechas escribiendo el ERS están basadas en información de los documentos de la propuesta del proyecto y en Requerimientos del usuario. El conjunto de requerimientos especificados en el ERS deben ser satisfechos en el diseño del sistema. Cualquier requisito funcional o no funcional que no sea identificado en el ERS, no debe aparecer en el producto final.

Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.0.0	06/12/11	Darwin Martínez	Versión inicial del artefacto
1.1.0	06/02/12	Darwin Martínez	Actualización del diagrama de casos de uso

Índice de Contenido

1	Introducción	5
	1.1 Alcance.....	5
	1.2 Definiciones, Siglas y Abreviaturas.....	5
	1.3 Referencias.....	5
2	Descripción General de la Solución.....	6
	2.1 Funciones del Producto.....	6
	2.2 Características del Usuario.....	6
3	Requerimientos Funcionales	6
	3.1 Matriz de Requerimientos.....	7
	3.2 Casos de Uso.....	7
	3.2.1 Resumen de Casos de Uso y Actores.....	7
	3.2.2 Diagrama General de Caso de Uso.....	8
	3.2.3 Descripción o Especificaciones de Casos de Uso.....	9
4	Requerimientos No Funcionales	14
	4.1 Usabilidad.....	14
	4.2 Confiabilidad.....	15
	4.3 Mantenimiento y Actualización	15
	4.4 Soportabilidad y Operabilidad.....	15
5	Anexos.....	15
	5.1 Prototipos.....	15

Especificación de Requerimientos del Software

1 Introducción

1.1 Alcance

Esta especificación de requerimientos está dirigida al usuario del sistema, así como a las personas que puedan utilizar las funcionalidades ya implementadas para continuar con el desarrollo de la aplicación. Dicha especificación presenta una descripción de los requerimientos funcionales (casos de uso definidos, los actores, las asociaciones entre estos, entre otros) y no funcionales (usabilidad, eficiencia, entre otros).

1.2 Definiciones, Siglas y Abreviaturas

- Azimut: Es el ángulo de una dirección contado en el sentido de las agujas del reloj a partir del norte geográfico
- Canal: es un objeto donde se almacena las muestras tomadas del pozo. Se presenta como una secuencia de datos indexados con respecto a la profundidad, el tiempo o alguna otra dimensión física del pozo. Estos datos vienen expresados como una matriz, de tamaño Mx1 en el caso de las curvas y MxN en caso de las imágenes.
- CU: Caso de uso.
- DLIS: *Digital Log Interchange Standard*, es la práctica recomendada número sesenta y seis (RP66) por el Instituto Americano del Petróleo la cual propone un estándar sintáctico para sísmica, perforación y perfilaje de pozos (<http://w3.energistics.org/rp66/v1/Toc/main.html>).
- Pad: es el objeto con el que la herramienta de perfilaje resistiva toma sus muestras. Las herramientas de perfilaje resistiva pueden tener cuatro, seis u ocho pad.
- Registro: información extraída o calculada a partir de los datos del archivo de entrada, es representado como una columna rectangular y posee información relevante del pozo. Existen registro de imagen (ver Figura 6), registro de curvas (ver Figura 7) y registros de imagen resistiva (ver Figura 8)

1.3 Referencias

Título	Fecha	Organización	Identificador del documento
Doc 01 - Descripción del Sistema.doc	26/07/10	PDVSA - Intevep	--

Título	Fecha	Organización	Identificador del documento
Doc 03 - Plugin Lector DLIS v1.doc	26/07/10	PDVSA - Intevep	--
Doc 04 - Plugin Modelado de Estructuras Geológicas.doc	26/07/10	PDVSA - Intevep	--
Doc 05 - Plugin Inteligencia en Deformaciones del Pozo.doc	27/07/10	PDVSA - Intevep	--
Doc 07 - Plugin Procesamiento de Registros.doc	27/07/10	PDVSA - Intevep	--
Doc 08 - Plugin Procesamiento Registros de Imágenes.doc	27/07/10	PDVSA - Intevep	--

2 Descripción General de la Solución

2.1 Funciones del Producto

Interpretar archivos de tipo DLIS, para generar registros de imágenes, dicho registro, tiene como principio físico la interpretación de una onda, bien sea acústica o resistiva, y por medio de este se puede generar un modelo de estabilidad del pozo, también a partir de los archivos de entrada se pueden generar curvas, con las cuales se hacen otros análisis, como la presión de poro y la presión de fractura, entre otros.

2.2 Características del Usuario

Ingeniero en petróleo o geólogos relacionados a la pericia de geomecánica/estabilidad de hoyo

3 Requerimientos Funcionales

- Leer archivo
- Generar curva
- Generar imagen
- Generar imagen resistiva
- Cambiar color de la curva
- Cambiar paleta de colores de la imágenes
- Hacer zoom (imágenes y curvas)

- Ver cuadrícula (sólo en el caso de las curvas)

3.1 Matriz de Requerimientos

Nro Req	Definición	Relacionado al módulo	Prioridad	Tipo	Status	Solicitado por	Analista receptor
REQ-01	Interpretar DLIS	Lectura de registro	E	F	IR	Carolina Zambrano	Darwin Martínez
REQ-02	Construir imagen	Generar imagen	E	F	IR	Carolina Zambrano	Darwin Martínez
REQ-03	Construir curva	Generar curva	E	F	IR	Carolina Zambrano	Darwin Martínez
REQ-04	Construir imagen resistiva	Generar imagen resistiva	E	F	IR	Carolina Zambrano	Darwin Martínez
REQ-05	Cambiar color	Generar curva	E	F	IR	Juan Ramos	Darwin Martínez
REQ-06	Ver cuadrícula	Generar curva	E	F	IR	Juan Ramos	Darwin Martínez
REQ-07	Cambiar paleta	Generar imagen / Generar imagen resistiva	E	F	IR	Carolina Zambrano	Darwin Martínez
REQ-08	Zoom	Generar imagen / Generar imagen resistiva / Generar curva	E	F	IR	Juan Ramos	Darwin Martínez

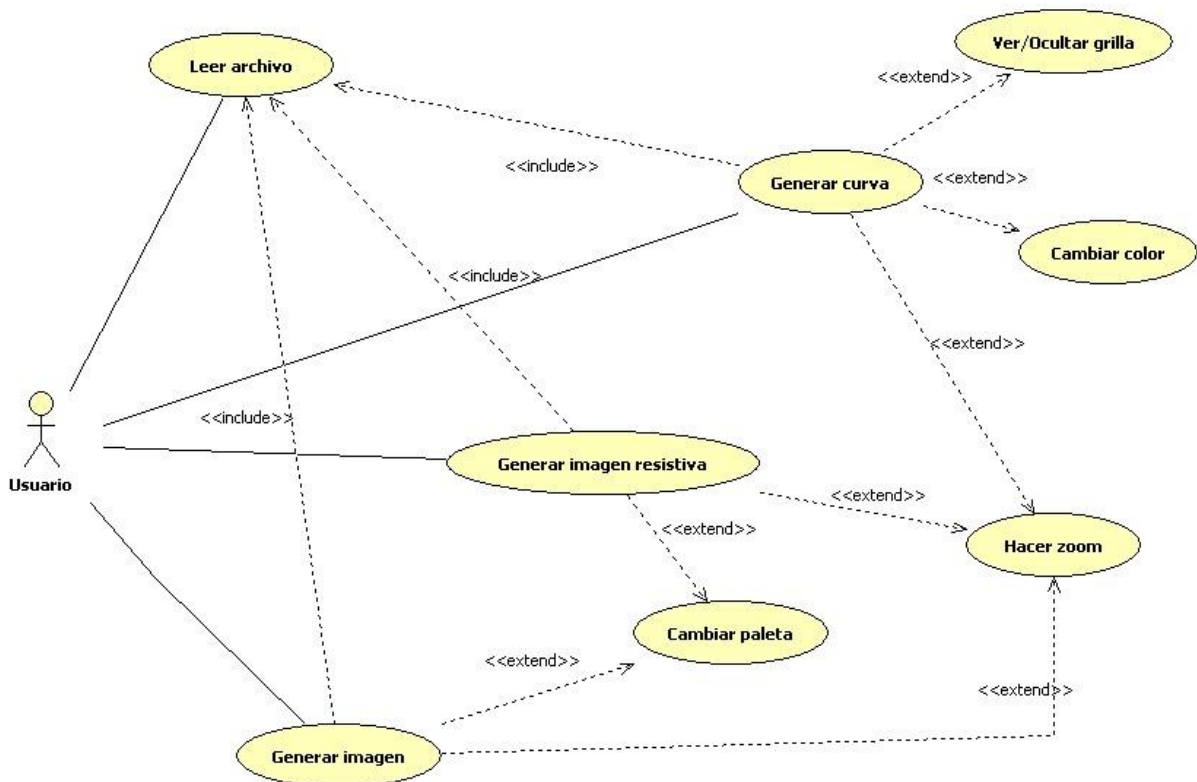
3.2 Casos de Uso

3.2.1 Resumen de Casos de Uso y Actores

Código	Caso de Uso	Actores participantes
001	Leer archivo	Usuario

Código	Caso de Uso	Actores participantes
002	Generar curva	Usuario
003	Generar imagen	Usuario
004	Generar imagen resistiva	Usuario
005	Cambiar color	Usuario
006	Ver/ocultar grilla	Usuario
007	Cambiar paleta	Usuario
008	Hacer zoom	Usuario

3.2.2 Diagrama General de Caso de Uso



3.2.3 Descripción o Especificaciones de Casos de Uso

Caso de Uso-001	
Nombre:	Leer archivo
Descripción:	Es la funcionalidad que permite interpretar un archivo DLIS y mostrar los datos relevantes para el usuario.
Requerimiento:	Interpretar DLIS
Precondición:	
Flujo Normal:	
Actor	Sistema
6. Introduce la ruta donde se encuentra el archivo DLIS.	7. Valida la ruta 8. Verifica que el archivo tenga una estructura valida. 9. Realiza la interpretación 10. Muestra los datos al usuario.
Flujo Alterno 1:	
Actor	Sistema
2.1. Introduce la ruta donde se encuentra el archivo DLIS.	2.2 Valida la ruta 2.3 Manda mensaje de ruta inválida.
Flujo Alterno 2:	
Actor	Sistema
2.5 Introduce la ruta donde se encuentra el archivo DLIS.	2.6 Valida la ruta. 2.7 Verifica que el archivo tenga una estructura valida. 2.8 Manda mensaje de archivo dañado.
Postcondición:	
Requerimientos Especiales:	.Adicionalmente esta funcionalidad permite exportar el archivo a formato HTML para que pueda ser legible para el usuario.
Puntos de Extensión:	
Puntos de Inclusión	<ul style="list-style-type: none"> • Generar imagen • Generar imagen resistiva • Generar curva

Caso de Uso-002	
Nombre:	Generar curva
Descripción:	Es la funcionalidad que permite construir una curva a partir de una matriz de datos de tamaño Mx1.
Requerimiento:	Construir curva
Precondición:	Haber ejecutado el CU Leer archivo
Flujo Normal:	
Actor	Sistema
7. Selecciona el (los) canal(es) que desea graficar 8. Envía la petición al sistema	9. Valida que el tamaño de la(s) matriz(es) de datos sea(n) de Mx1 10. Traduce el conjunto de bytes al tipo de dato asociado al (los) canal(es). 11. Construye la(s) curva(s). 12. Muestra la(s) curva(s) al usuario.
Flujo Alternativo 1:	
Actor	Sistema
3.1. Selecciona el(los) canal(es) que desea graficar. 3.2 Envía la petición al sistema	3.3 Valida que el tamaño de la(s) matriz(es) de datos sea(n) de Mx1. 3.4 Manda mensaje de tamaño inválido.
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	<ul style="list-style-type: none"> • Cambiar color • Ver/Ocultar grilla • Hacer zoom
Puntos de Inclusión	

Caso de Uso-003	
Nombre:	Generar imagen
Descripción:	Es la funcionalidad que permite construir una imagen a partir de una matriz de datos de tamaño MxN.
Requerimiento:	Construir imagen
Precondición:	Haber ejecutado el CU Leer archivo
Flujo Normal:	
Actor	Sistema
7. Selecciona el(los) canal(es) que desea graficar. 8. Envía la petición al sistema	9. Valida que el tamaño de la(s) matriz(es) de datos sea(n) de MxN 10. Traduce el conjunto de bytes al tipo de dato

Caso de Uso-003	
	asociado al (los) canal(es). 11. Construye la(s) imagen(es). 12. Muestra la(s) imagen(es) al usuario.
Flujo Alterno 1:	
Actor	Sistema
4.1. Selecciona el(los) canal(es) que desea graficar. 4.2 Envía la petición al sistema	4.3 Valida que el tamaño de la(s) matriz(es) de datos sea(n) de MxN. 4.4 Manda mensaje de tamaño inválido.
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	<ul style="list-style-type: none"> • Cambiar paleta • Hacer zoom
Puntos de Inclusión	

Caso de Uso-004	
Nombre:	Generar imagen resistiva
Descripción:	Es la funcionalidad que permite construir una imagen resistiva a partir de la curva de azimut y cuatro, seis u ocho canales de imágenes equivalentes a los pad de la herramienta de perfilaje resistivo.
Requerimiento:	Construir imagen resistiva
Precondición:	Haber ejecutado el CU Leer archivo
Flujo Normal:	
Actor	Sistema
1. Selecciona los canales correspondiente a la imagen resistiva 2. Envía la petición al sistema	3. Valida que solo exista un canal de curva (azimut) 4. Valida que los canales de imagen solo sean 4, 6 u 8. 5. Valida que las matrices de datos asociadas a los canales de imagen sean todas del mismo tamaño. 6. Construye la imagen resistiva. 7. Muestra la imagen al usuario.
Flujo Alterno 1:	
Actor	Sistema
1.1. Selecciona los canales correspondientes a la imagen resistiva.	1.3 Realiza la validación de los canales 1.4 Manda mensaje error en los canales seleccionados.

Caso de Uso-004	
1.2 Envía la petición al sistema	
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	<ul style="list-style-type: none"> Cambiar paleta Hacer zoom
Puntos de Inclusión	

Caso de Uso-005	
Nombre:	Cambiar color
Descripción:	Es la funcionalidad que permite cambiar de color a las curvas generadas.
Requerimiento:	Cambiar color
Precondición:	Haber ejecutado el CU Generar curva
Flujo Normal:	
Actor	Sistema
1. Selecciona la curva que se desea cambiar el color 2. Envía la petición al sistema 4. Seleccionar color	3. Mostrar cuadro de dialogo con los colores disponibles. 5. Cambiar de color a la curva.
Flujo Alterno 1:	
Actor	Sistema
1.1. Selecciona la curva que se desea cambiar el color 1.2 Envía la petición al sistema 1.4 Cancelar petición.	1.3 Mostrar cuadro de dialogo con los colores disponibles.
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	
Puntos de Inclusión	

Caso de Uso-006	
Nombre:	Ver/Ocultar grilla
Descripción:	Es la funcionalidad que permite mostrar u ocultar una cuadrícula en las curvas generadas

Caso de Uso-006	
Requerimiento:	Ver cuadrícula
Precondición:	Haber ejecutado el CU Generar curva
Flujo Normal:	
Actor	Sistema
1. Selecciona la curva en la cual se desea ver/ocultar la cuadrícula 2. Envía la petición al sistema	3. Mostrar /ocultar la cuadrícula en la curva seleccionada.
Flujo Alternativo 1:	
Actor	Sistema
Postcondición:	
Requerimientos Especiales:	.
Puntos de Extensión:	
Puntos de Inclusión	

Caso de Uso-007	
Nombre:	Cambiar paleta
Descripción:	Es la funcionalidad que permite cambiar la tonalidad de las imágenes generadas.
Requerimiento:	Cambiar paleta
Precondición:	Haber ejecutado el CU Generar imagen o Generar imagen resistiva
Flujo Normal:	
Actor	Sistema
1. Selecciona la imagen a la que se desea cambiar la tonalidad 2. Envía la petición al sistema	3. Aplica la siguiente paleta de color disponible de la lista.
Flujo Alternativo 1:	
Actor	Sistema
Postcondición:	
Requerimientos Especiales:	.

Caso de Uso-007	
Puntos de Extensión:	
Puntos de Inclusión	

Caso de Uso-008	
Nombre:	Hacer zoom
Descripción:	Es la funcionalidad que permite ampliar o reducir la escala vertical de las imágenes o curvas generadas.
Requerimiento:	Zoom
Precondición:	Haber ejecutado el CU Generar imagen, Generar imagen resistiva y/o Generar curva
Flujo Normal:	
Actor	Sistema
1. Selecciona la imagen o curva a la que se desea aumentar la escala 2. Envía la petición al sistema	3. El sistema redibuja la imagen o curva ampliando la escala vertical en un 25% de su tamaño actual
Flujo Alterno 1:	
Actor	Sistema
1. Selecciona la imagen o curva a la que se desea reducir la escala 2. Envía la petición al sistema	3. El sistema redibuja la imagen o curva reduciendo la escala vertical en un 25% de su tamaño actual
Postcondición:	
Requerimientos Especiales:	Adicionalmente esta funcionalidad permite restaurar la imagen o curva a su escala original.
Puntos de Extensión:	
Puntos de Inclusión	

4 Requerimientos No Funcionales

4.1 Usabilidad

El software será de fácil uso, ya que las interfaces estarán diseñadas pensando en las necesidades del usuario con iconos alusivos y textos de ayuda que faciliten la interacción con el programa, por lo que el tiempo de aprendizaje será mínimo.

4.2 Confiabilidad

El software será confiable, ya que, asegura un nivel de funcionamiento adecuado bajo condiciones normales, y de recuperarse manteniendo su nivel de funcionamiento en caso de errores.

4.3 Mantenimiento y Actualización

El sistema será construido de forma modular, con alta cohesión y bajo acoplamiento entre módulos, por lo que el mantenimiento y actualización será una tarea sencilla ya que los cambios hechos en una modulo no afectaran a otro.

4.4 Soportabilidad y Operabilidad

El sistema cumplirá con Soportabilidad ya que este puede ser brindado por el grupo de desarrolladores de la Suite de Perforación. Será operable ya que el software permitirá al usuario operarlo y controlarlo de una manera amigable e intuitiva.

5 Anexos

5.1 Prototipos

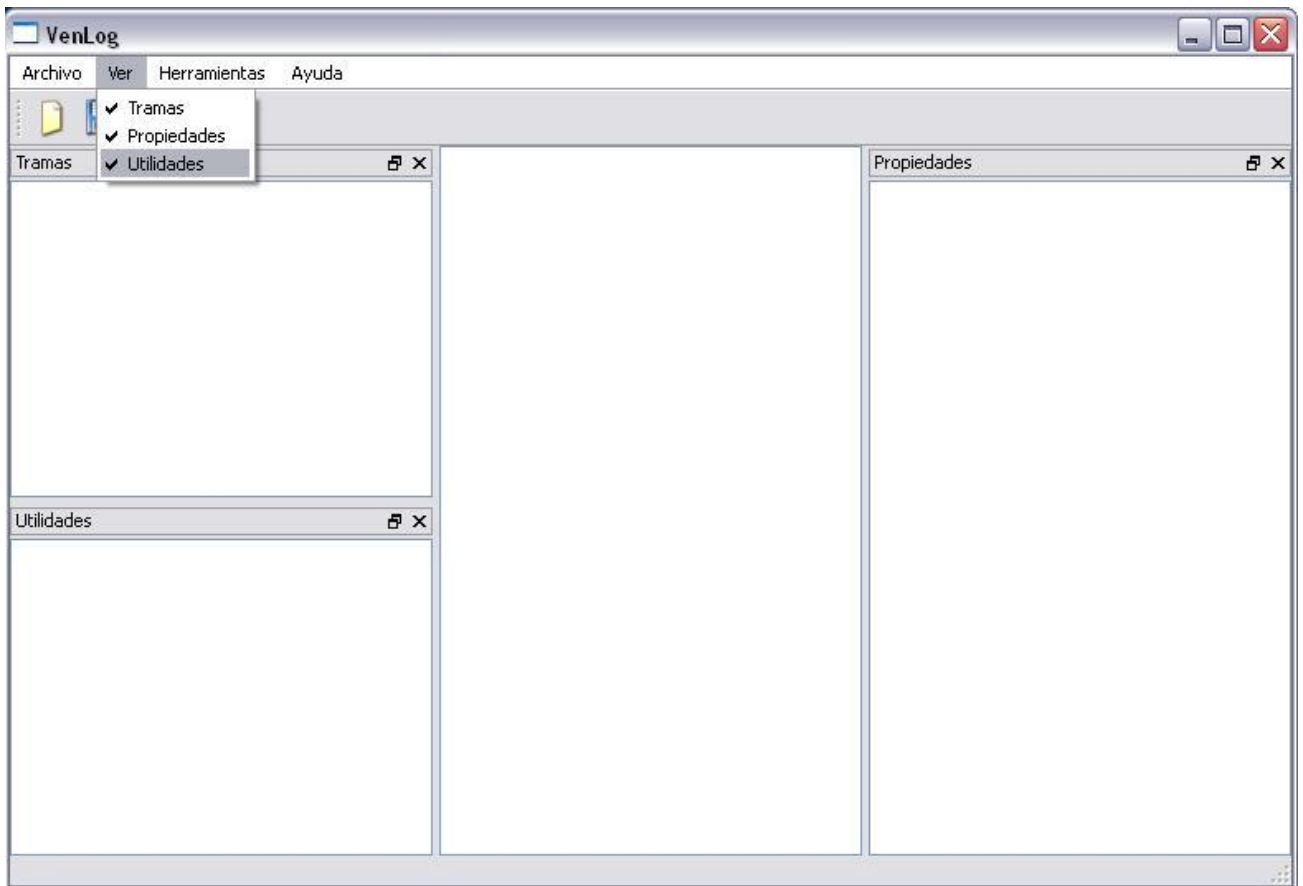


Figura 1. Pantalla inicial del sistema
Fuente: Darwin Martínez

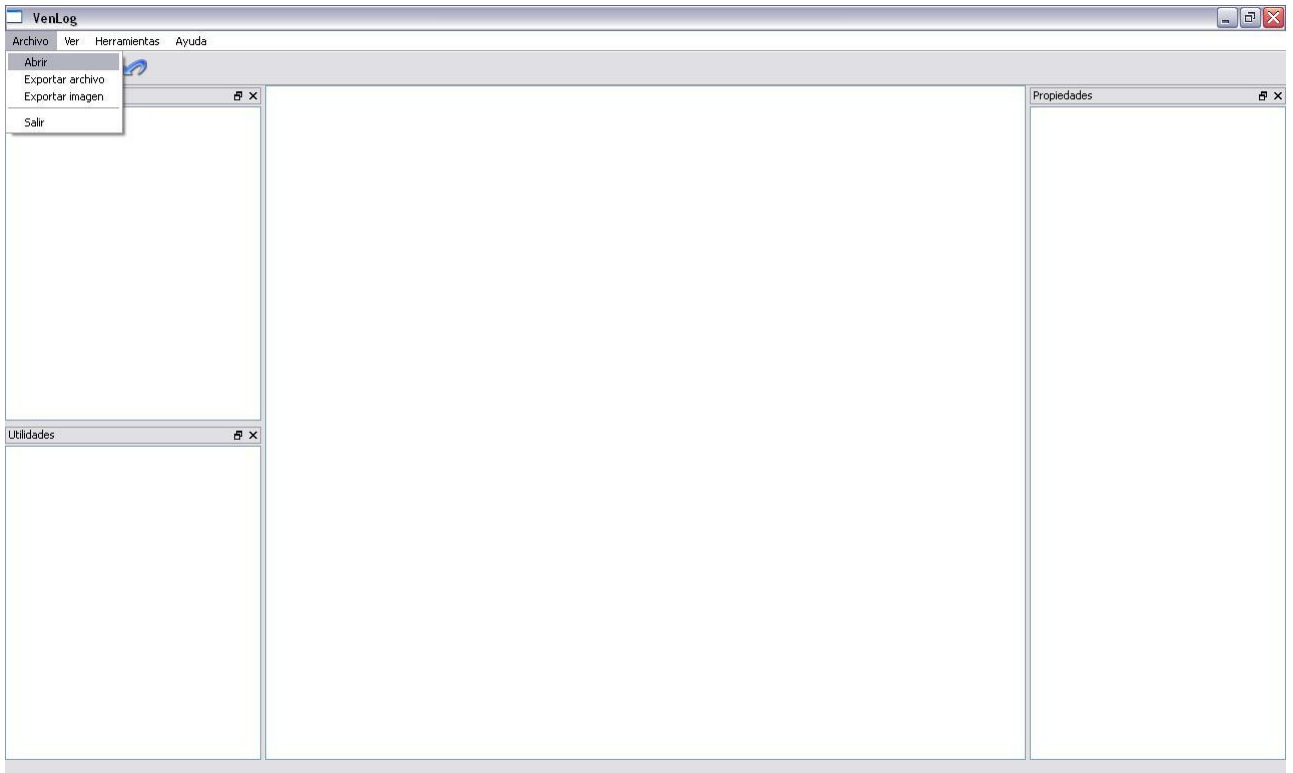


Figura 2. Menú archivo
Fuente: Darwin Martinez

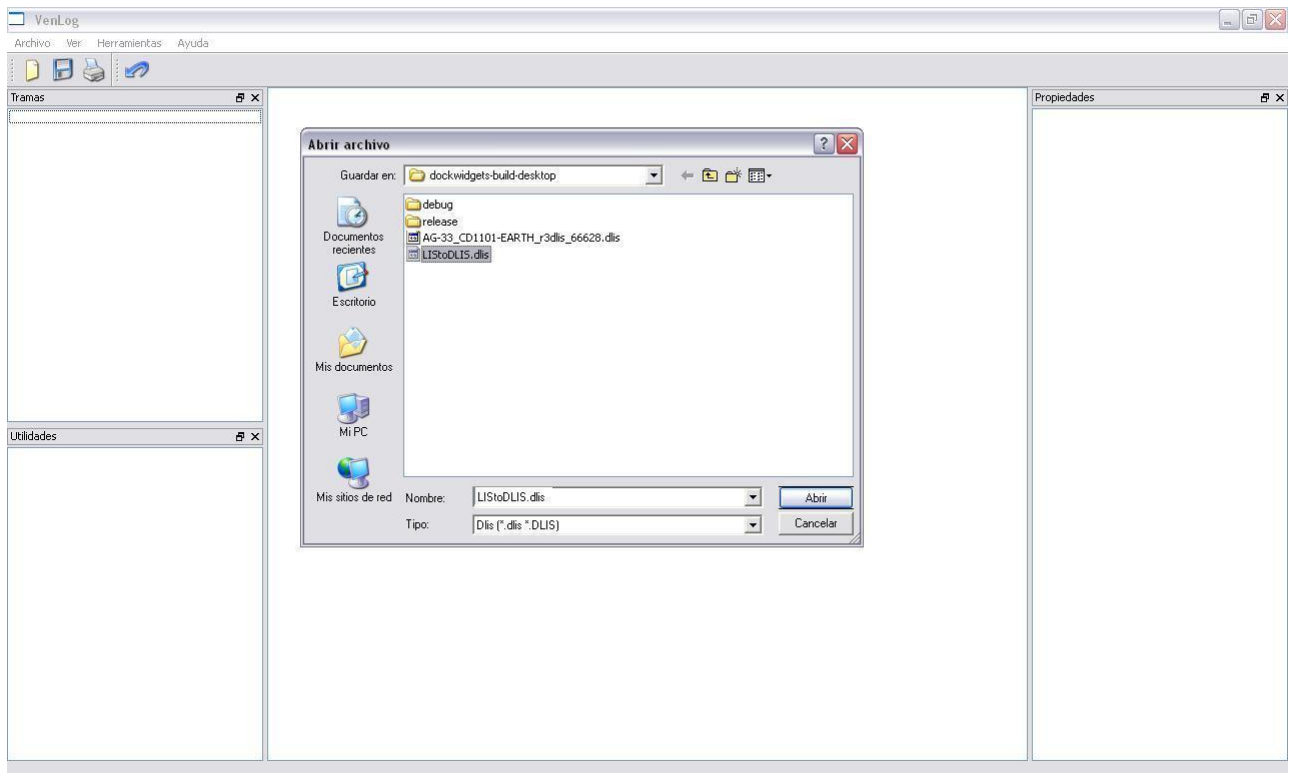
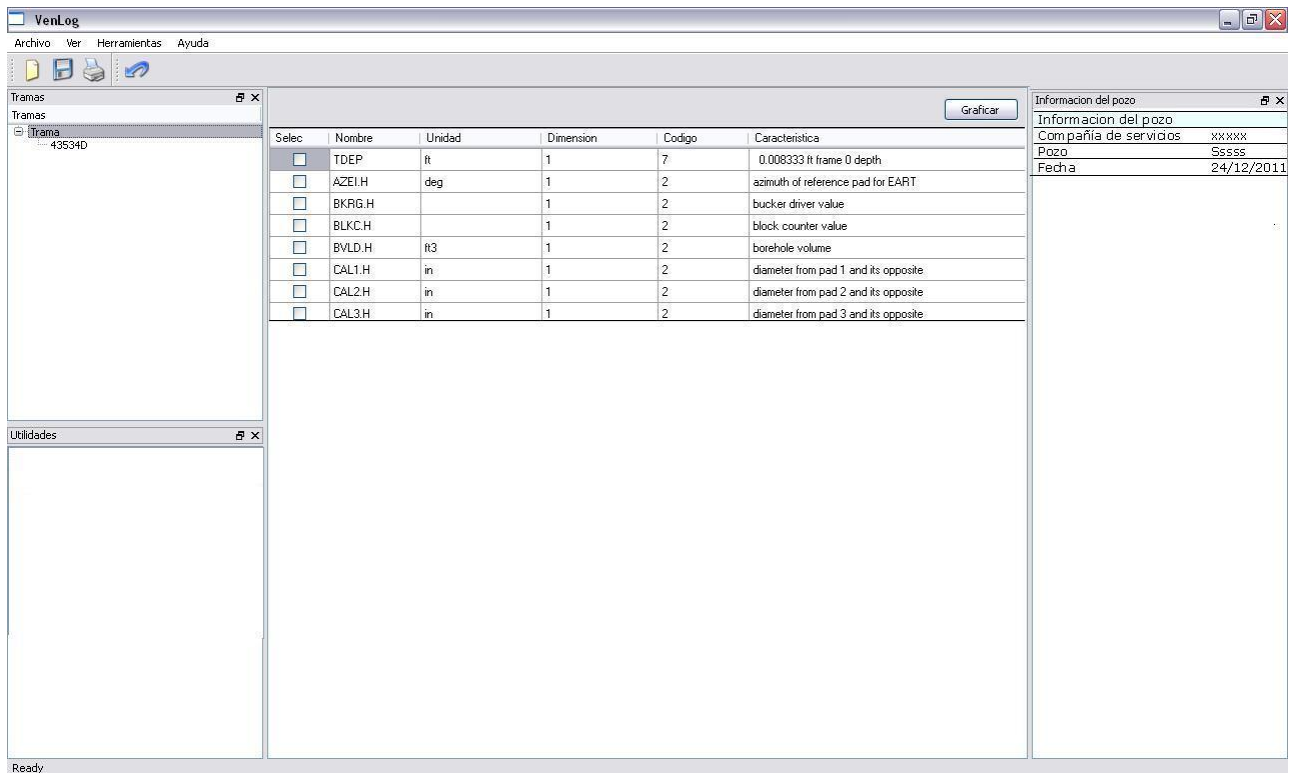


Figura 3. Abrir archivo
Fuente: Darwin Martinez



The screenshot shows the VenLog software interface. The main window displays a table with the following data:

Selec	Nombre	Unidad	Dimension	Codigo	Caracteristica
<input type="checkbox"/>	TDEP	ft	1	7	0.008333 ft frame 0 depth
<input type="checkbox"/>	AZEL.H	deg	1	2	azimuth of reference pad for EART
<input type="checkbox"/>	BKRG.H		1	2	bucket driver value
<input type="checkbox"/>	BLKC.H		1	2	block counter value
<input type="checkbox"/>	BVLD.H	ft3	1	2	borehole volume
<input type="checkbox"/>	CAL1.H	in	1	2	diameter from pad 1 and its opposite
<input type="checkbox"/>	CAL2.H	in	1	2	diameter from pad 2 and its opposite
<input type="checkbox"/>	CAL3.H	in	1	2	diameter from pad 3 and its opposite

Additional interface elements include a 'Tramas' panel on the left with a tree view showing 'Trama' and '43534D', a 'Utilidades' panel at the bottom left, and an 'Informacion del pozo' panel on the right with the following data:

Informacion del pozo	
Compañía de servicios	xxxxx
Pozo	S5555
Fecha	24/12/2011

Figura 4. Archivo interpretado

Fuente: Darwin Martínez

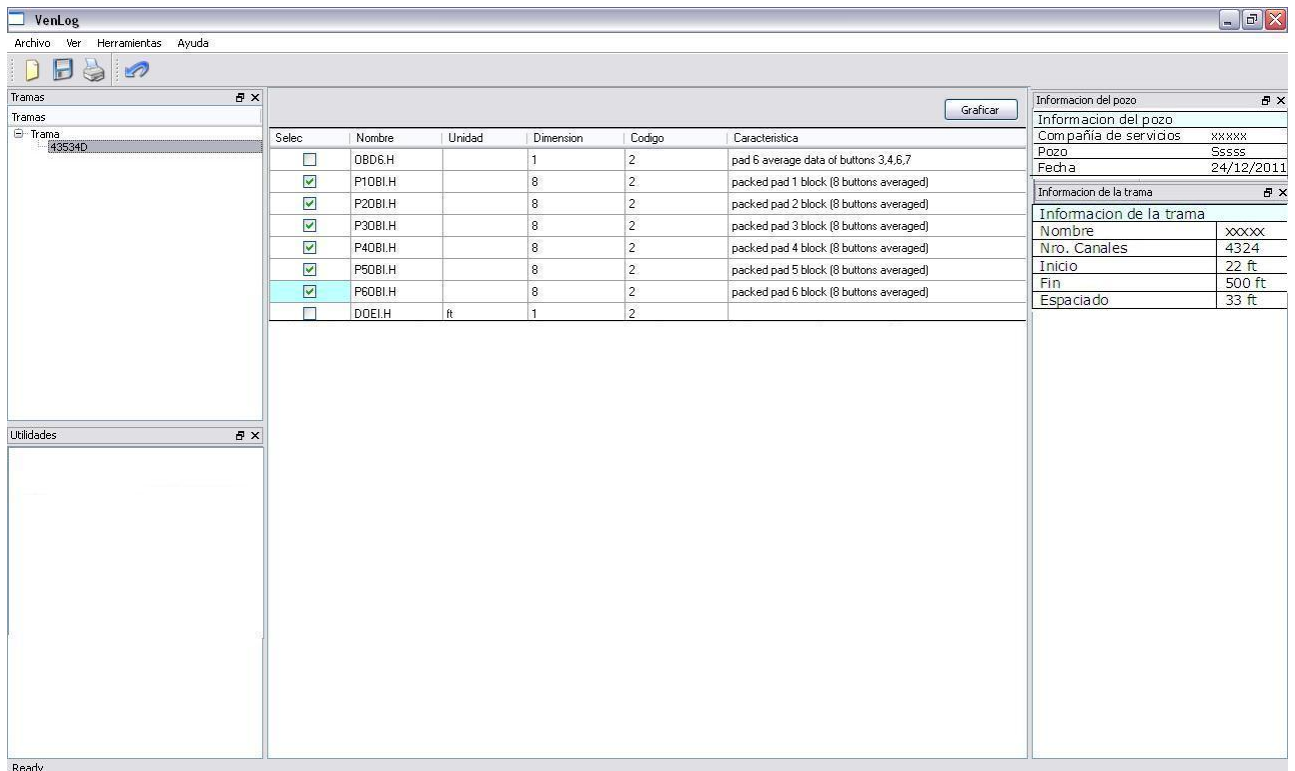


Figura 5. Selección de canales de imágenes

Fuente: Darwin Martínez

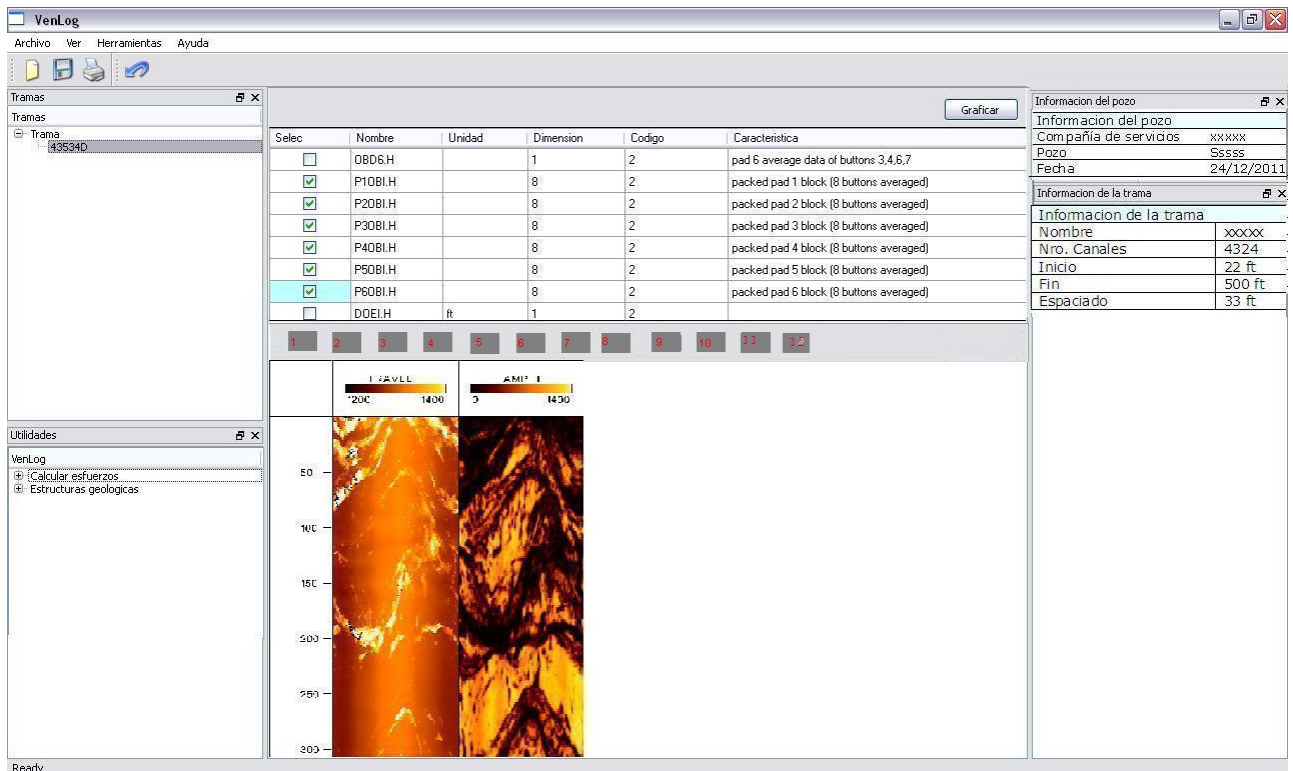


Figura 6. Imagen generada a partir de la selección de canales

Fuente: Darwin Martínez

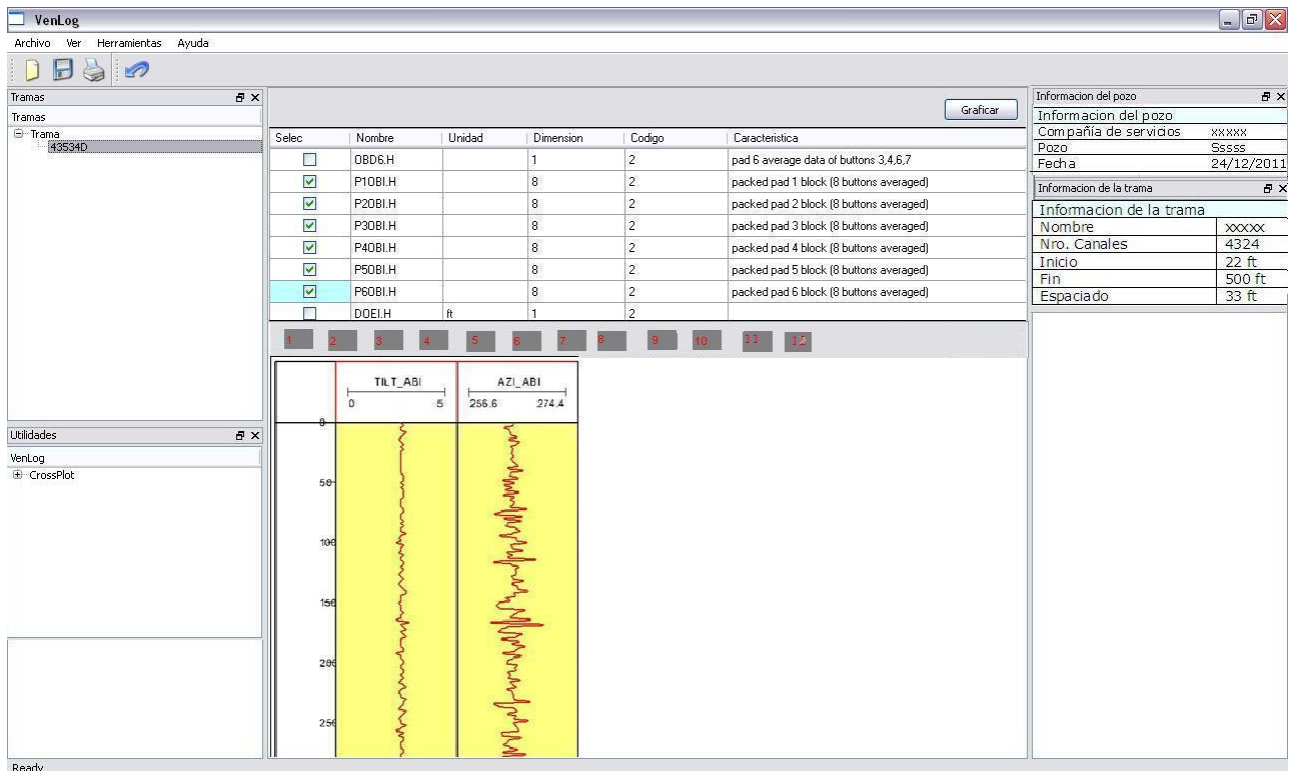


Figura 7. Registros de curvas generados
 Fuente: Darwin Martinez

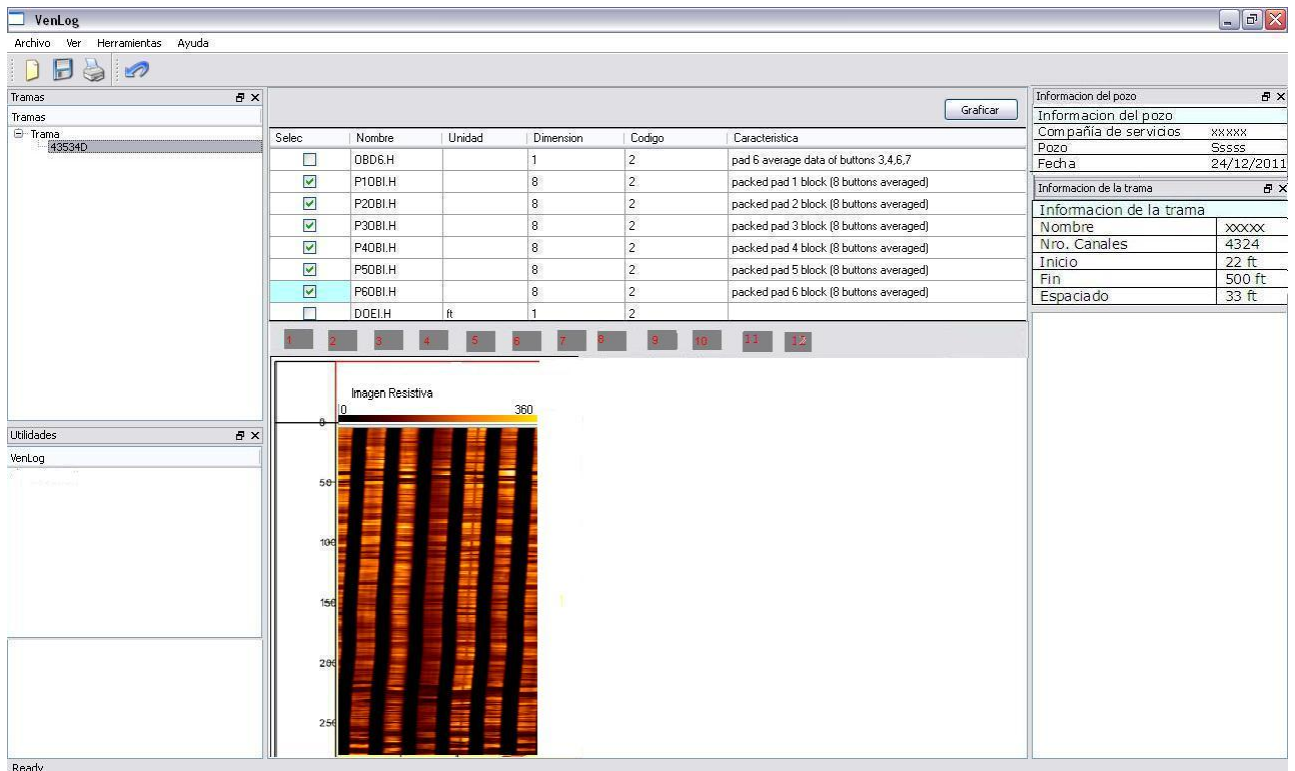


Figura 8. Registros de imagen resistiva generado
 Fuente: Darwin Martínez

Copyright 2011 PDVSA – Gerencia Funcional AIT. Bajo licencia de Documentación Libre de GNU

Copyright 2009 PDVSA Intevep. Bajo licencia de Documentación Libre de GNU

Copyright (C) 2007 CNTI. Todos los derechos reservados.

Copyright (C) 2003 Jason Robbins. Todos los derechos reservados.

Copyright (C) 2006 CNTI. Todos los derechos reservados.

Anexo 2



Gerencia AIT
(Automatización, Informática y
Telecomunicaciones)

Documento de Arquitectura del Software

< VenLog 2.0 >

Versión: < 1.1.0 >

< Octubre, 2012 >

Descripción del Documento

Es una especificación de las ideas principales del diseño. El DAS proporciona una descripción entendible de la arquitectura del sistema software y sirve como medio de comunicación entre el arquitecto de software y otros miembros de equipo del proyecto con respecto a las decisiones arquitectónicamente significativas que se han tomado en el proyecto. Contiene varias vistas que muestran aspectos distintos del sistema como son: Vista de Casos de Uso, Vista Lógica, Vista de Implementación, Vista del Proceso, Vista de Implantación y Vista de Datos.

Las vistas involucradas en el Documento de Arquitectura del Software (DAS) se detallan a continuación.

Vista de Casos de Uso: Esta vista muestra la funcionalidad del sistema como es percibida desde el exterior. Así como también describe un conjunto de escenarios y casos de uso que tienen una cobertura arquitectónicamente significativa o que ilustran un punto específico de la arquitectura. Es un subconjunto del Modelo de Casos de Uso y además su realización es obligatoria.

Vista Lógica: Describe el diseño más importante de las clases y su organización en paquetes y subsistemas, y la organización de éstos en capas. También contiene algunas realizaciones de casos de uso. Esta muestra cómo la funcionalidad es diseñada en el interior del sistema, en términos de la estructura estática y comportamiento dinámico del sistema. Es un subconjunto del Modelo de Casos de Uso y su realización es obligatoria.

Vista de Implementación: Esta vista muestra la organización del código y el código actual de ejecución. Contiene una visión general del Modelo de Implementación y su organización en términos de módulos en paquetes y capas. También se describe la asignación de paquetes y clases de la Vista Lógica a los paquetes y módulos de la Vista de Implementación. Es un subconjunto del Modelo de Implementación.



Esta vista es opcional, ya que sólo se realiza en los casos donde la implementación no se conduce estrictamente por el diseño. Si el empaquetado de los modelos de diseño y de implementación son idénticos, esta vista puede ser omitida.

Vista de Procesos: En esta vista se describe las tareas, sus interacciones y configuraciones, y la asignación de objetos del diseño y clases a las tareas. Muestra los elementos relacionados con el desempeño incluyendo escalabilidad, concurrencia y tiempo base de desempeño. Es un subconjunto del Modelo de Diseño y se usa sólo si el sistema tiene un grado significativo de concurrencia, por lo tanto es una vista opcional.

Vista de Implantación: Describe varios nodos físicos para las configuraciones más típicas de las plataformas y la asignación de las tareas de la Vista del Proceso a los nodos físicos. Es un subconjunto del Modelo de Implantación. Esta vista se realiza sólo si el sistema es distribuido a través de más de un nodo, por lo tanto es opcional.

Vista de Datos: Esta vista especifica arquitectónicamente los elementos constantes en el Modelo de Datos. Describe una apreciación global del modelo de los datos y su organización por lo que se refiere a las tablas, vistas y almacenamiento de los procedimientos que proporcionan la persistencia al sistema. También describe la cartografía de clases constantes de la Vista Lógica a la estructura de los datos de la base de datos.

Esta vista es opcional, ya que sólo se realiza si la persistencia es un aspecto significativo del sistema y el traslado del Modelo de Diseño al Modelo de Datos no se hace automáticamente por el mecanismo de persistencia.



Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.0.0	23/04/2012	Darwin Martínez	Documento de Arquitectura versión inicial
1.1.0	28/08/2012	Darwin Martínez	Actualización del diagrama de clases



Índice de Contenido

1	Introducción	97
	1.1 Alcance.....	97
	1.2 Definiciones, Acrónimos y Abreviaturas.....	97
2	Resumen Arquitectónico	97
	2.1 Hechos más Importantes.....	97
	2.2 Estilo Arquitectónico.....	98
	2.3 Objetivos de la Arquitectura.....	98
3	Componentes Significativos de la Arquitectura del Sistema	98
3.1	Componente 1: C-01 Framework Qt.....	98
3.2	Componente 2: C-02 Librería CImg.....	99
3.3	Componente 3: C-03 Gráficos 2D.....	99
3.4	Componente 4: C-04 VisualizadorDLIS	99
3.5	Componente 5: C-05 VenLogIU	100
3.6	Componente 6: C-06 LectorDLIS.....	100
4	Vista de Casos de Uso	100
5	Vista Lógica	101
6	Vista de Implementación	107
7	Integración.....	108
	7.1 Integración de los Componentes y su Comunicación.....	108
	7.2 Mecanismos de la Arquitectura para Futuras Modificaciones o Extensiones.....	108
8	Lista de Control de la Arquitectura	108
9	Aseguramiento de la Calidad.....	109
	9.1 Alcance del Plan de Calidad.....	109
	9.2 Objetivos de Calidad.....	109



Documento de Arquitectura del Software

1 Introducción

1.1 Alcance

En este documento se describe la arquitectura propuesta para el proyecto VenLog 2.0 en su módulo lector de registros, por parte del equipo Fábrica de Software de AIT Miranda Oeste.

1.2 Definiciones, Acrónimos y Abreviaturas

C++: es un lenguaje de programación diseñado por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido (<http://www.cplusplus.com/info/history/>).

Qt: es un framework multiplataforma usado para desarrollar aplicaciones con una interfaz gráfica de usuario así como también para el desarrollo de programas sin interfaz gráfica como herramientas para la línea de comandos y consolas para servidores (<http://qt.digia.com/product/>)

Cimg: es una librería de software de código abierto, escrita en C++, disponible gratuitamente para realizar gráficos en 2D y 3D, procesamiento de imágenes y la visualización. (<http://cimg.sourceforge.net/>).

DLIS: *Digital Log Interchange Standard*, es la práctica recomendada número sesenta y seis (RP66) por el Instituto Americano del Petróleo la cual propone un estándar sintáctico para sísmica, perforación y perfilaje de pozos. (<http://w3.energistics.org/rp66/v1/Toc/main.html>).

2 Resumen Arquitectónico

2.1 Hechos más Importantes

El motivo principal que da origen a esta solución es la necesidad de los ingenieros y científicos especialistas en estabilidad de hoyos de PDVSA – Intevep, de contar con una herramienta que les permita analizar el estado del hoyo desnudo a través de las imágenes que proporciona la herramienta de lectura de registros acústicos y resistivos. Es por esto que se decidió realizar un lector de archivos DLIS que además de brindar independencia para interpretar el registro y realizar análisis de rupturas, fracturas naturales e inducidas, entre otros eventos geológicos, permita ampliar la matriz de conocimiento y abrir líneas de investigación en esta materia.



2.2 Estilo Arquitectónico

El estilo de arquitectura utilizado en el desarrollo de VenLog 2.0, módulo lector de registros, es una arquitectura dividida en 2 capas que separa los datos de la aplicación en presentación (interfaz de usuario) y lógica de control.

La capa **lógica de control**, define la lógica de negocio y es la que se encarga de hacer la interpretación de los archivos DLIS.

La capa de **presentación**, es la encargada de la navegabilidad del sistema, la presentación de los datos, interfaz gráfica de usuario y la validación de los datos de entrada.

2.3 Objetivos de la Arquitectura

Los principales objetivos de la arquitectura propuesta son:

Reutilización, los servicios y componentes desarrollados deben ser consumibles y utilizables por las aplicaciones que se desarrollen, incurriendo en una disminución de tiempo y esfuerzo al momento del desarrollo así como en la estandarización de los mismos.

Funcionalidad, debe proveer la funcionalidad adecuada para la consecución de los objetivos, adecuándose a los requerimientos funcionales y no funcionales, prestando especial atención a los objetivos de calidad presentados en el punto 9 de este documento.

Portabilidad, debido a la migración progresiva de sistemas Windows a Linux por parte de la empresa, la solución planteada debe ser capaz de correr en ambas plataformas.

3 Componentes Significativos de la Arquitectura del Sistema

3.1 Componente 1: C-01 Framework Qt

C-01: Framework Qt	
Descripción:	Framework que permite la construcción de interfaces de usuario.
Requerimientos de software:	Compilador genérico de c++ (GCC en el caso de distribuciones Linux y MinGW en el caso de Windows).
Requerimientos de hardware:	1Gb de RAM, procesador AMD Sempron 130 u otro procesador con características similares o superiores.
Requerimientos de plataforma	Windows: XP, Vista, 7. o distribuciones Linux.

3.2 Componente 2: C-02 Librería CImg

C-02: Librería CImg	
Descripción:	Librería que permite la creación y manipulación de imágenes en 2D
Requerimientos de software:	Compilador genérico de c++ (GCC en el caso de distribuciones Linux y MinGW en el caso de Windows).
Requerimientos de hardware:	1Gb de RAM, procesador AMD Sempron 130 u otro procesador con características similares o superiores.
Requerimientos de plataforma	Windows: XP, Vista, 7. o distribuciones Linux.

3.3 Componente 3: C-03 Gráficos 2D

C-03: Gráficos 2D	
Descripción:	Componente que permite realizar graficas (imágenes o curvas) de los datos contenidos en los archivos de entrada, específicamente en los canales seleccionados por el usuario.
Requerimientos de software:	Compilador genérico de c++ (GCC en el caso de distribuciones Linux y MinGW en el caso de Windows).
Requerimientos de hardware:	1Gb de RAM, procesador AMD Sempron 130 u otro procesador con características similares o superiores.
Requerimientos de plataforma	Windows: XP, Vista, 7. o distribuciones Linux.

3.4 Componente 4: C-04 VisualizadorDLIS

C-04: VizualizadorDLIS	
Descripción:	Componente que muestra los datos contenidos en un archivo de tipo DLIS.
Requerimientos de software:	Compilador genérico de c++ (GCC en el caso de distribuciones Linux y MinGW en el caso de Windows).
Requerimientos de hardware:	1Gb de RAM, procesador AMD Sempron 130 u otro procesador con características similares o superiores.
Requerimientos de plataforma	Windows: XP, Vista, 7. o distribuciones Linux.

3.5 Componente 5: C-05 VenLogIU

C-05: VenLogIU	
Descripción:	Componente que permite la visualización y graficación de los datos obtenidos de un archivo de tipo DLIS por medio de los componentes anteriores.
Requerimientos de software:	Compilador genérico de c++ (GCC en el caso de distribuciones Linux y MinGW en el caso de Windows).
Requerimientos de hardware:	1Gb de RAM, procesador AMD Sempron 130 u otro procesador con características similares o superiores.
Requerimientos de plataforma	Windows: XP, Vista, 7. o distribuciones Linux.

3.6 Componente 6: C-06 LectorDLIS

C-06: LectorDlis	
Descripción:	Interpreta los datos almacenados en archivos de tipo DLIS.
Requerimientos de software:	Compilador genérico de c++ (GCC en el caso de distribuciones Linux y MinGW en el caso de Windows).
Requerimientos de hardware:	1Gb de RAM, procesador AMD Sempron 130 u otro procesador con características similares o superiores.
Requerimientos de plataforma	Windows: XP, Vista, 7. o distribuciones Linux.

4 Vista de Casos de Uso

A continuación se presenta el diagrama general de casos de usos para los paquetes mencionados anteriormente. La razón por la cual el diagrama de casos de uso en la fase de requerimientos tiene más funcionalidades, es porque durante el levantamiento de requerimientos se tomaron en cuenta todas las funciones que debe realizar el sistema final, mientras que en esta fase sólo se tomaron en cuenta las funciones que atañen únicamente al módulo lector de registros.

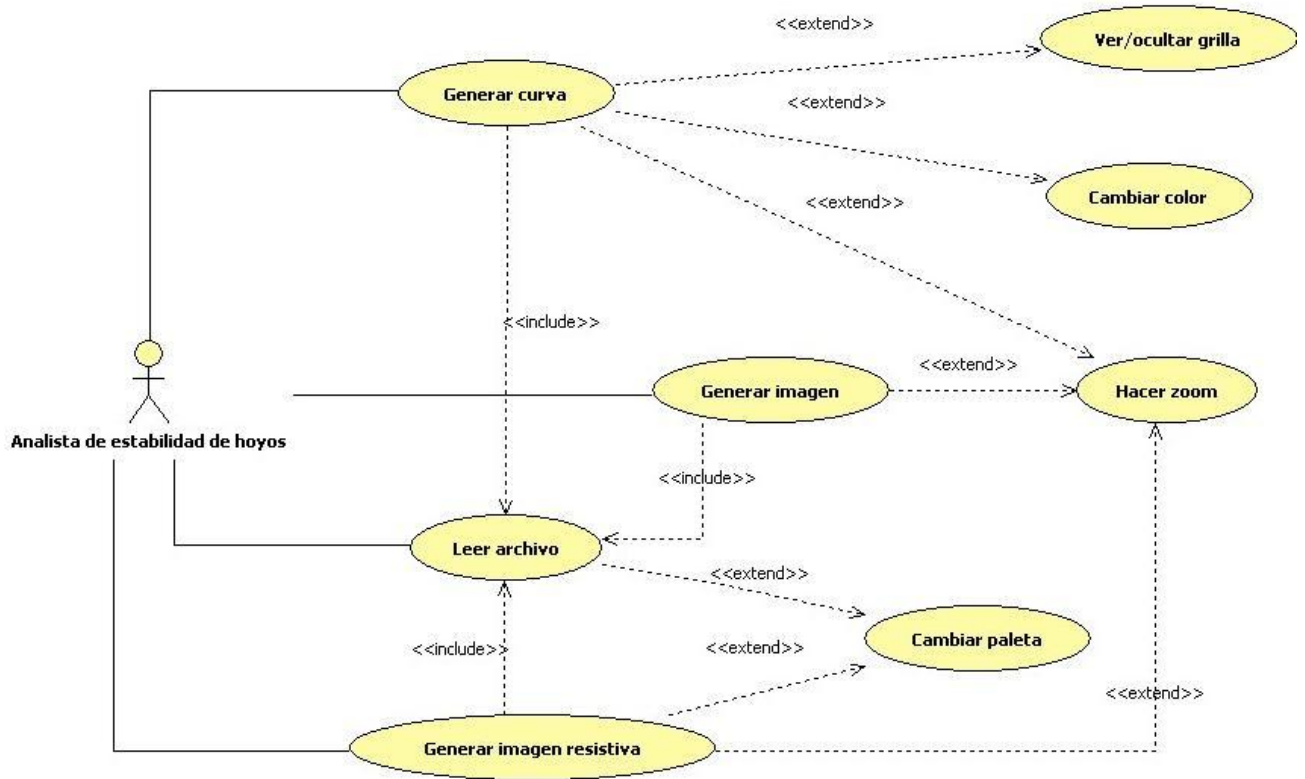


Figura 1. Diagrama General de Casos de Usos

NOTA: La especificación de estos casos de uso se encuentra en el documento ERS

5 Vista Lógica

A continuación se presenta un diagrama de clases que representa el contenido de los componentes del sistema, se observan los parámetros y métodos de cada clase, así como las relaciones entre ellas.

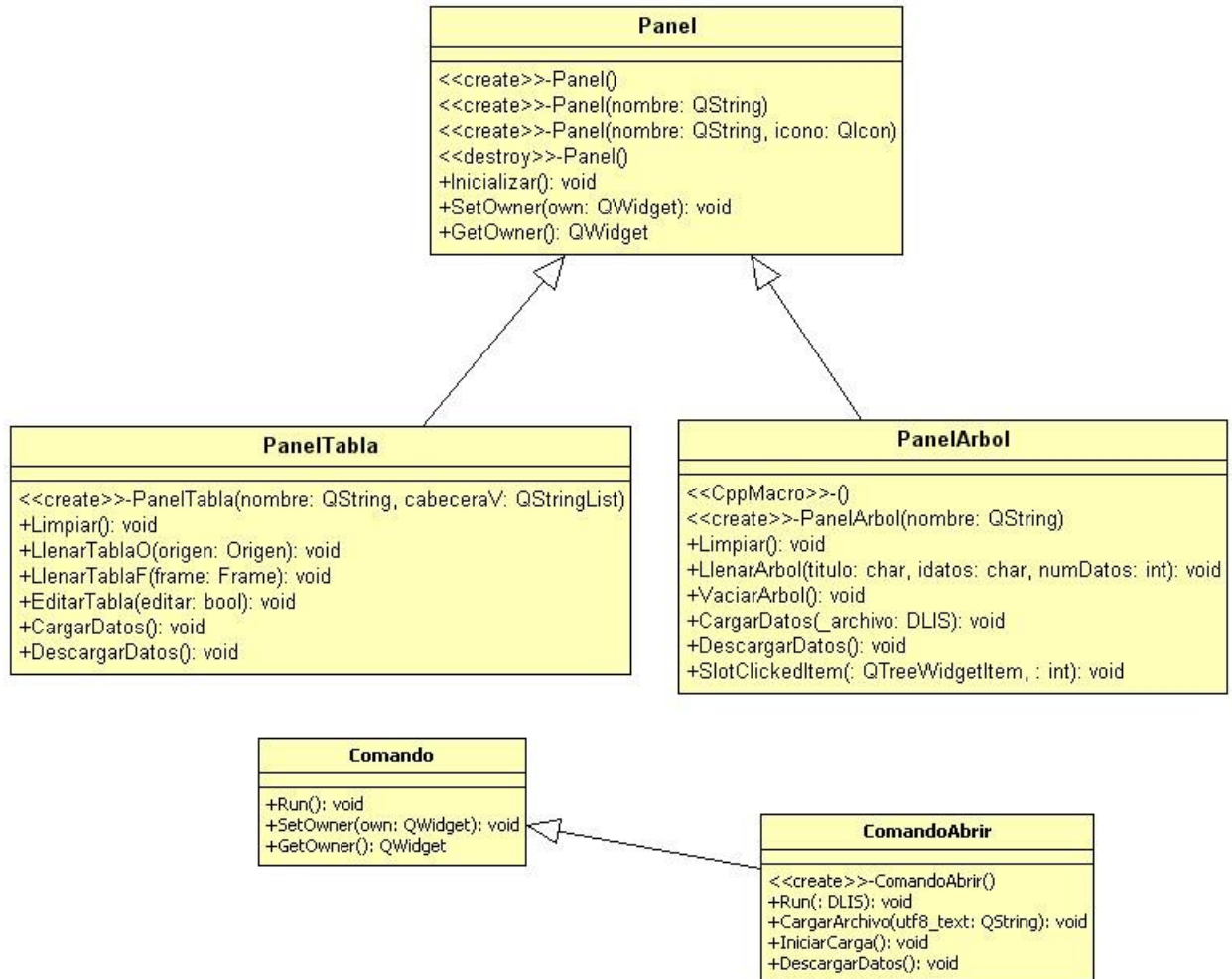


Figura 2-A. Diagrama de clase capa presentación



Figura 2-B. Diagrama de clase capa presentación

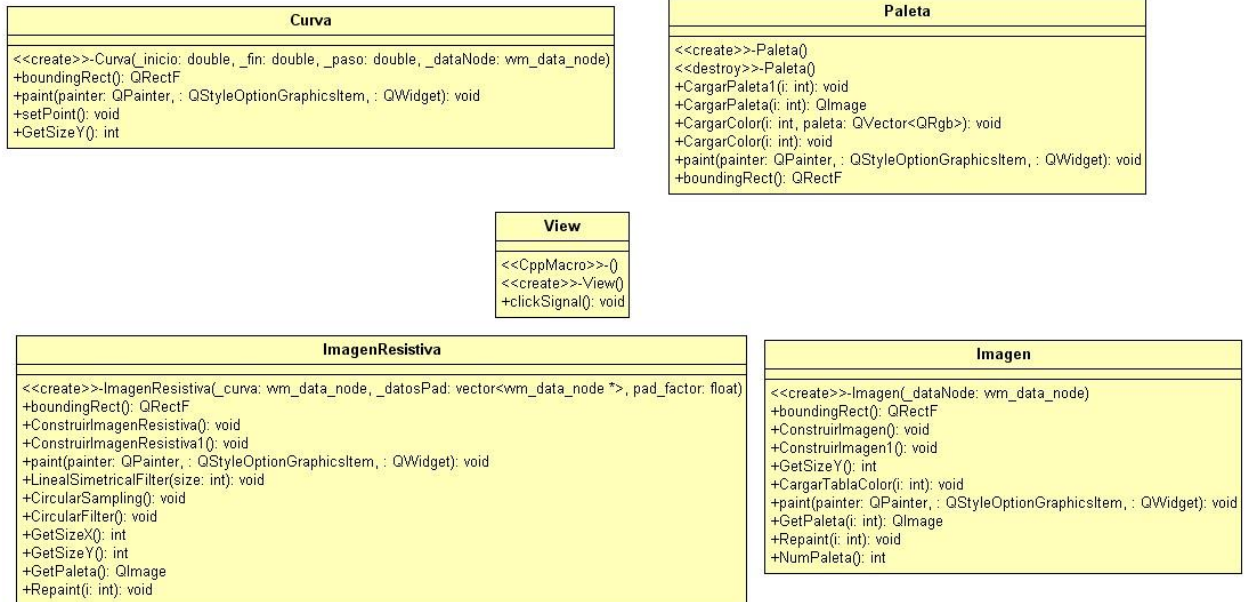


Figura 2-C. Diagrama de clase capa presentación

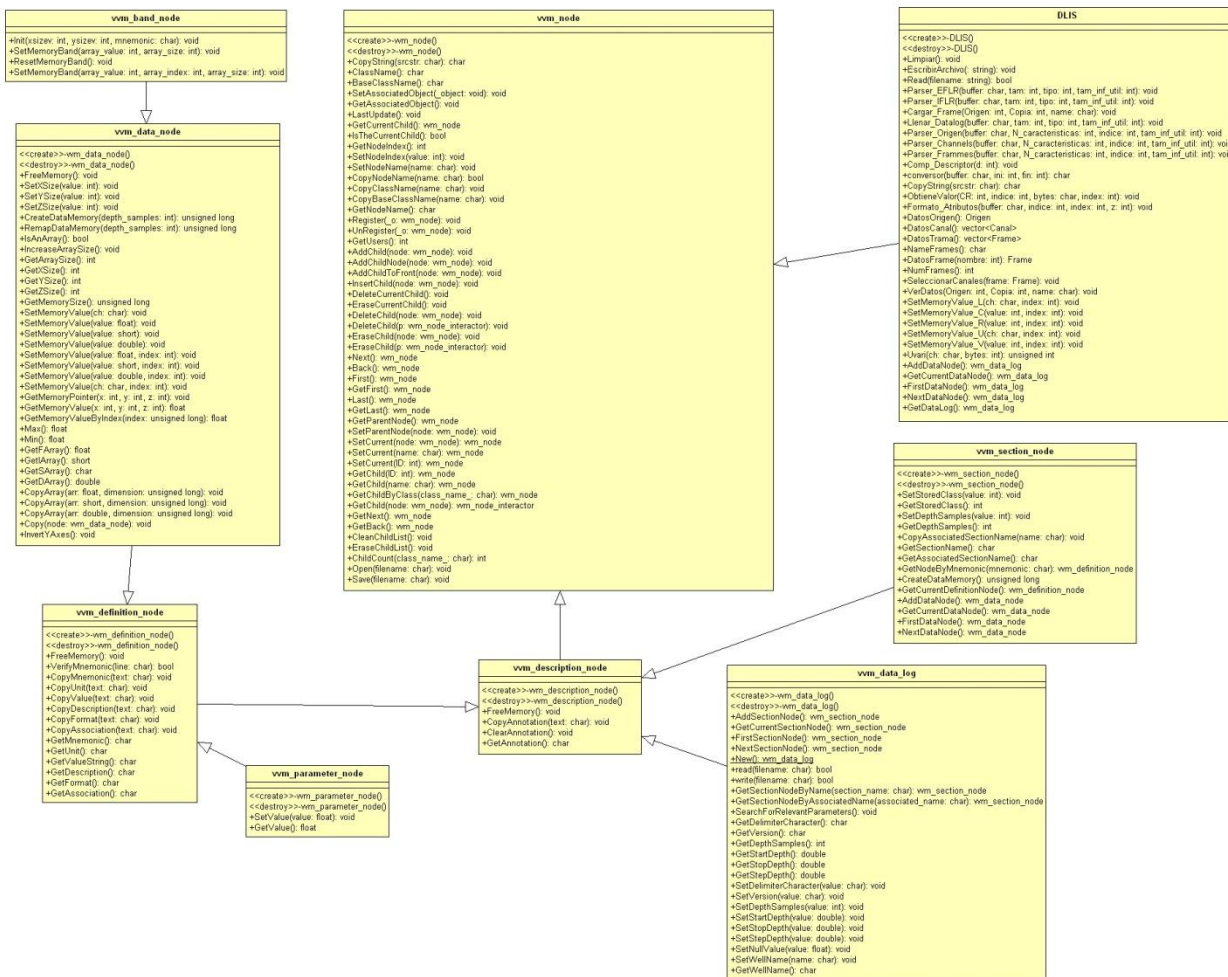


Figura 3A. Diagrama de clase capa lógica



< PRCP >
< VenLog 2.0 >

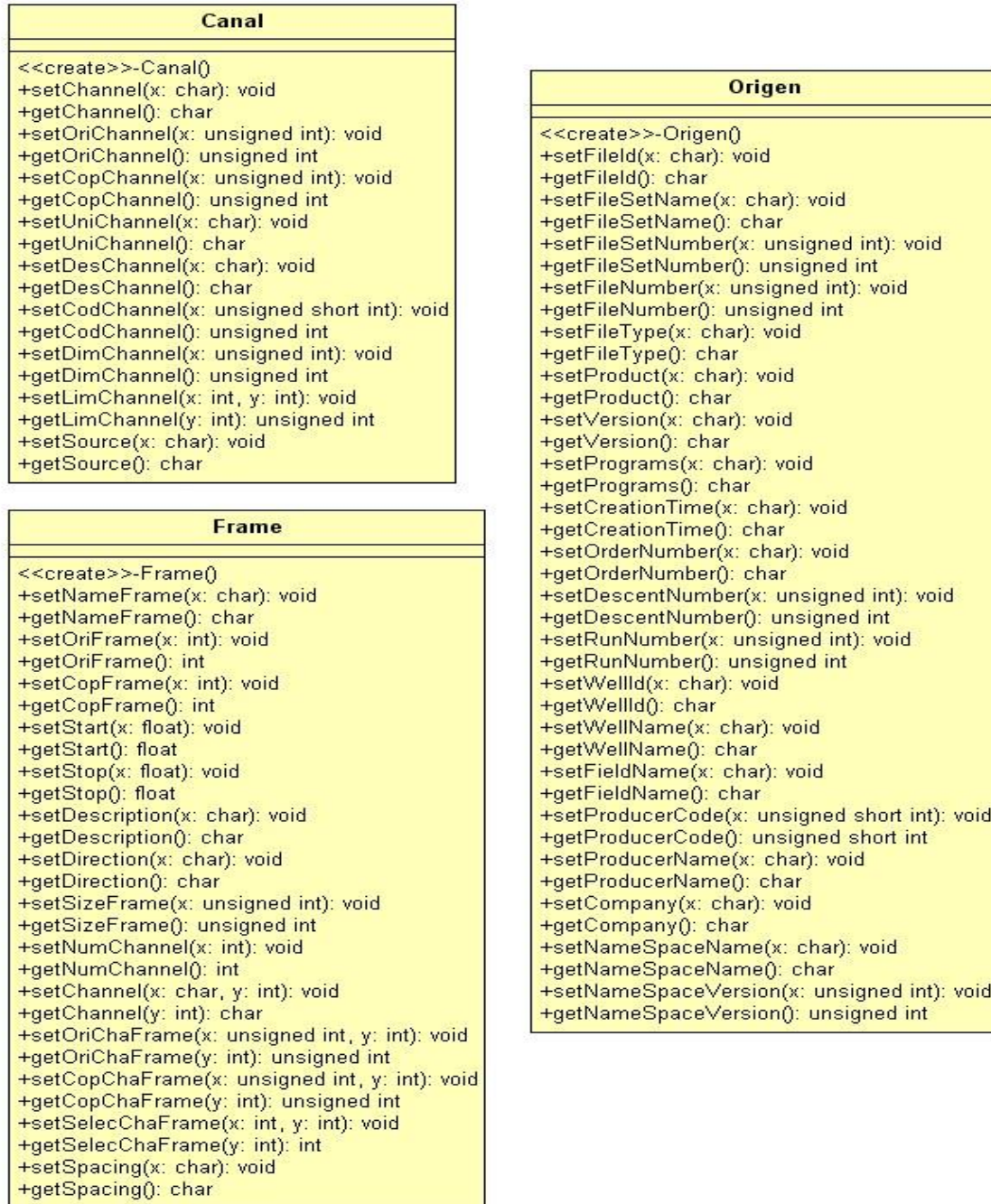


Figura 3B. Diagrama de clase capa lógica

6 Vista de Implementación

La implementación de los componentes correspondientes a VenLog 2.0, módulo lector de registros, estarán divididos en 2 capas, la capa lógica, la cual se encarga de procesar la información contenida en los archivos de entrada y procesar las peticiones del usuario (ver Figura 4) y la capa de presentación la cual se encargará de la representación de la información y de la interacción con el usuario.

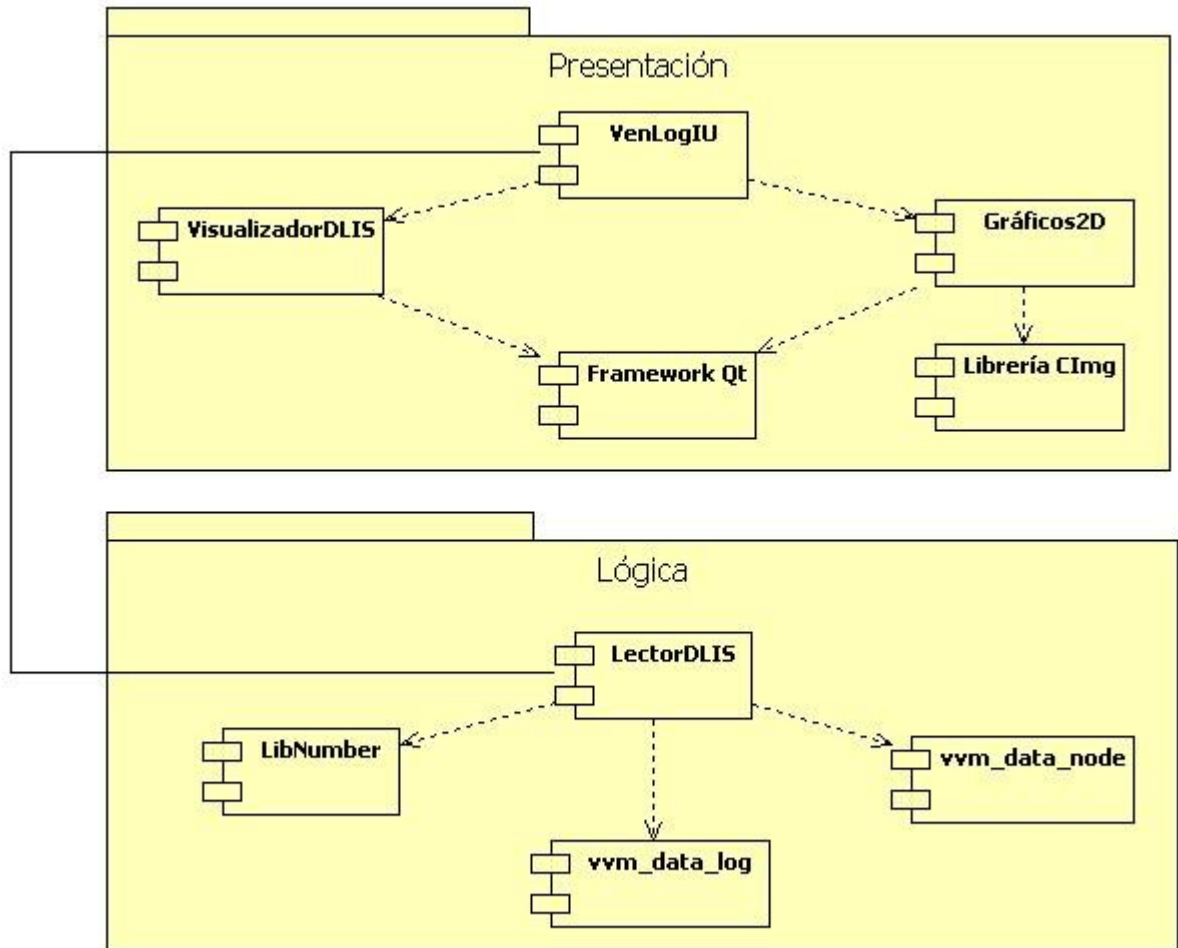


Figura 4. Diagrama de componentes VenLog 2.0



7 Integración

7.1 Integración de los Componentes y su Comunicación

La integración de los componentes se hará por medio de llamadas directas a procedimientos.

7.2 Mecanismos de la Arquitectura para Futuras Modificaciones o Extensiones

Dado que el sistema será construido en forma modular, las extensiones o nuevos componentes podrán ser cargados sin ningún problema. Las modificaciones que se realicen en un modulo existente no afectaran a los otros módulos relacionados con este, mientras no varíen los parámetros de entrada y/o de salida, de ser así, será necesario hacer un ajuste en los módulos relacionados.

8 Lista de Control de la Arquitectura

Facilidad de Integración

En este sistema, todos los componentes están diseñados para trabajar juntos. Y los componentes nuevos o reutilizados son integrados con interfaces simples.

Expansibilidad

Este sistema se puede expandir añadiendo las nuevas funcionalidades como módulos o componentes independientes que se comunicaran con los existentes mediante interfaces, de tal modo que los componentes existentes consuman los servicios de los nuevos y viceversa.

Consistencia y Familiaridad

Los usuarios podrán aplicar sus conocimientos de interfaces similares o interfaces estándar a este sistema.

Corrección

El diseño se ajusta correctamente a los requerimientos dados.



Viabilidad

Este diseño puede ser implementado y probado con las cantidades de tiempo y esfuerzo planeadas.

9 Aseguramiento de la Calidad

9.1 Alcance del Plan de Calidad

Los aspectos a ser evaluados en el plan de calidad de la arquitectura son los que se describen a continuación. El alcance de los mismos corresponde a los componentes desarrollados por el equipo de trabajo.

9.2 Objetivos de Calidad

De acuerdo con los lineamientos del proyecto se presentan los siguientes objetivos de calidad agrupados según su prioridad:

9.2.1 Esenciales

- Funcionalidad > Corrección
- Funcionalidad > Robustez
- Usabilidad > Comprensibilidad y Legibilidad

9.2.2 Esperados

- Funcionalidad > Exactitud
- Funcionalidad > Corrección medible
- Usabilidad > Comprensibilidad y Legibilidad
- Usabilidad > Eficiencia
- Usabilidad > Consistencia y Familiaridad
- Usabilidad > Satisfacción Subjetiva

9.2.3 Deseados

- Confiabilidad > Consistencia en carga
- Capacidad de mantenimiento > Comprensibilidad
- Capacidad de mantenimiento > Capacidad de evolución
- Capacidad de mantenimiento > Capacidad de prueba



PDVSA

Gerencia AIT
(Automatización, Informática y Telecomunicaciones)
Especificación de Requisitos del Software

< PRCP >
< VenLog 2.0 >

Versión
PDV-SGAIIT-
DAS-001
Página 110 de
147

Copyright 2011 PDVSA – Gerencia Funcional AIT. Bajo licencia de Documentación Libre de GNU

Copyright 2009 PDVSA Intevep. Bajo licencia de Documentación Libre de GNU

Copyright (C) 2007 CNTI. Todos los derechos reservados.

Copyright (C) 2003 Jason Robbins. Todos los derechos reservados.

Copyright (C) 2006 CNTI. Todos los derechos reservados.

Anexo 3

Publicado en gaceta oficial N° 38.095 de fecha 28/12/2004

Decreto 3390

Hugo Chaves Frías

Presidente de la República

De conformidad con lo expuesto en los artículos 110 y 226 de la Constitución de la República Bolivariana de Venezuela, 12 y 47 de la Ley Orgánica de la Administración Pública y 2, 19 y 22 de Decreto de Rango y Fuerza de la Ley Orgánica de Ciencia, Tecnología e Innovación, en Consejo de Ministros

... Decreta:

Artículo 1. *La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado en estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollados con estándares abiertos.*

Artículo 2. *A los efectos del presente decreto se entenderá por:*

Software Libre: *Programa de computación cuya licencia garantiza al usuario acceso al código fuente del programa y lo autoriza a ejecutarlo con cualquier propósito, modificarlo y redistribuir tanto el programa original como sus modificaciones en las mismas condiciones de licenciamiento acordadas al programa original, sin tener que pagar regalías a los desarrolladores previos.*

Estándares Abiertos: *Especificaciones técnicas, publicadas y controladas por alguna organización que se encarga de su desarrollo, las cuales han sido aceptadas por la industria, estando a disposición de cualquier usuario para ser implementadas en un software libre u otro, promoviendo la competitividad, interoperabilidad o flexibilidad.*

Software Propietario: *Programa de computación cuya licencia establece restricciones de uso, redistribución o modificación por parte de los usuarios, o requiere de autorización expresa del Licenciador.*

Distribución Software Libre desarrollado con Estándares Abiertos para el Estado Venezolano: Un paquete de programas y aplicaciones de Informática elaborado utilizando Software Libre con Estándares Abiertos para ser utilizados y distribuidos entre distintos usuarios.

Artículo 3. En los casos que no se puedan desarrollar o adquirir aplicaciones en Software Libre bajo Estándares Abiertos, los órganos y entes de la Administración Pública Nacional deberán solicitar ante el Ministerio de Ciencia y Tecnología autorización para adoptar otro tipo de soluciones bajo las normas y criterios establecidos por ese Ministerio.

Artículo 4. El Ministerio de Ciencia y Tecnología, adelantará los programas de capacitación de los funcionarios públicos, en el uso del Software Libre desarrollado con Estándares Abiertos, haciendo especial énfasis en los responsables de las áreas de tecnologías de información y comunicación, para lo cual establecerá con los demás órganos y entes de la Administración Pública Nacional los mecanismos que se requieran.

Artículo 5. El Ejecutivo Nacional fomentará la investigación y desarrollo de software bajo modelo Software Libre desarrollado con Estándares Abiertos, procurando incentivos especiales para desarrolladores.

Artículo 6. El Ejecutivo Nacional fortalecerá el desarrollo de la industria nacional del software, mediante el establecimiento de una red de formación, de servicios especializados en Software Libre desarrollado con Estándares Abiertos y desarrolladores.

Artículo 7. El Ministerio de Ciencia y Tecnología será responsable de proveer la Distribución Software Libre desarrollado con Estándares Abiertos para el Estado Venezolano, para lo cual implementará los mecanismos que se requieran.

Artículo 8. El Ejecutivo Nacional promoverá el uso generalizado del Software Libre desarrollado con Estándares Abiertos en la sociedad, para lo cual desarrollará mecanismos orientados a capacitar e instruir a los usuarios en la utilización del Software Libre desarrollado con Estándares Abiertos.

Artículo 9. El Ejecutivo Nacional promoverá la cooperación internacional en materia de Software Libre desarrollado con Estándares Abiertos, con especial énfasis en la

cooperación regional a través del MERCOSUR, CAN, CARICOM y la cooperación SUR-SUR.

Artículo 10. *El Ministerio de Educación y Deportes, en coordinación con el Ministerio de Ciencia y Tecnología, establecerá las políticas para incluir el Software Libre desarrollado con Estándares Abiertos, en los programas de educación básica y diversificada.*

Artículo 11. *En un plazo no mayor de noventa (90) días continuos, contados a partir de la publicación del presente Decreto en la Gaceta Oficial de la República Bolivariana de Venezuela, el Ministerio de Ciencia y Tecnología deberá presentar ante la Presidencia de la República, los planes y programas que servirán de plataforma para la ejecución progresiva del presente Decreto.*

Artículo 12. *Cada Ministro en coordinación con la Ministra de Ciencia y Tecnología, en un plazo no mayor de noventa (90) días continuos, contados a partir de la aprobación por parte de la Presidencia de la República de los planes y programas referidos en el artículo anterior, publicará en la Gaceta Oficial de la República Bolivariana de Venezuela su respectivo plan de implantación progresiva del Software Libre desarrollado con Estándares Abiertos, acogiéndose a los lineamientos contenidos en aquellos, incluyendo estudios de financiamiento e incentivos fiscales a quienes desarrollen Software Libre con Estándares Abiertos destinados a la aplicación de los objetivos previstos en el presente Decreto. Igualmente, las máximas autoridades de sus entes adscritos publicaran a través del Ministerio de adscripción sus respectivos planes.*

Los planes de implantación progresiva del Software Libre desarrollado con Estándares Abiertos de los distintos órganos y entes de la Administración Pública Nacional, deberán ejecutarse en un plazo no mayor de veinticuatro (24) meses, dependiendo de las características propias de sus sistemas de información. Los Ministros mediante Resolución y las máximas autoridades de los entes que le estén adscritos a través de sus respectivos actos, determinarán las fases de ejecución del referido Plan, así como las razones de índole técnico que imposibiliten la implantación progresiva del Software Libre en los casos excepcionales, de acuerdo a lo establecido en el artículo 3 del presente Decreto.

Artículo 13. *El Ministerio de Ciencia y Tecnología establecerá dentro de los planes y programas contemplados en el presente Decreto, mecanismos que preserven la identidad y necesidades culturales del país, incluyendo a sus grupos indígenas, para*

lo cual procurará que los sistemas operativos y aplicaciones que se desarrollen se adecuen a su cultura.

Artículo 14. *Todos los Ministros quedan encargados de la ejecución del presente Decreto, bajo la coordinación de la Ministra de Ciencia y Tecnología.*

Anexo 4

Nombre: Cristina Ramírez

Cargo: Analista de Mantenimiento AIT Miranda Oeste

Fecha: 16/10/2012

Por favor, valore los siguientes aspectos del sistema utilizando una escala del 1 al 5, donde 1 es "**En completo desacuerdo**" y 5 es "**Completamente de acuerdo**". Su respuesta ayudará a mejorar la usabilidad del sistema.

#	Preguntas	Valoración
1	Creo que me gustaría utilizar este sistema con frecuencia.	5
2	Encontré el sistema innecesariamente complejo	1
3	Creo que el sistema es fácil de usar.	5
4	Creo que voy a necesitar el apoyo de un técnico para poder utilizar este sistema.	1
5	Me pareció que las diversas funciones en este sistema están bien integradas.	5
6	Creo que hay inconsistencia en este sistema.	1
7	Me parece que la mayoría de los usuarios aprenderá a utilizar este sistema muy rápidamente.	5
8	Me pareció que el sistema es complicado de usar.	1
9	Me sentí seguro de usar el sistema.	5
10	Me parece que necesito aprender muchas cosas antes de poder interactuar con el sistema.	1

#	Preguntas adicionales	Valoración
11	Pude hacer la tarea que deseaba fácilmente	5
12	Encontré la información que buscaba	5
13	Obtuve la ayuda requerida al usar el sistema	5
14	Recomendaría usar este sistema	5

Nombre: Carmen R. Salazar G.

Cargo: Analista Fábrica de Software Miranda Oeste - Gerencia AIT - PDVSA

Fecha: 16/10/2012

Por favor, valore los siguientes aspectos del sistema utilizando una escala del 1 al 5, donde 1 es "**En completo desacuerdo**" y 5 es "**Completamente de acuerdo**". Su respuesta ayudará a mejorar la usabilidad del sistema.

#	Preguntas	Valoración
1	Creo que me gustaría utilizar este sistema con frecuencia.	5
2	Encontré el sistema innecesariamente complejo	1
3	Creo que el sistema es fácil de usar.	5
4	Creo que voy a necesitar el apoyo de un técnico para poder utilizar este sistema.	2
5	Me pareció que las diversas funciones en este sistema están bien integradas.	4
6	Creo que hay inconsistencia en este sistema.	1
7	Me parece que la mayoría de los usuarios aprenderá a utilizar este sistema muy rápidamente.	1
8	Me pareció que el sistema es complicado de usar.	4
9	Me sentí seguro de usar el sistema.	4
10	Me parece que necesito aprender muchas cosas antes de poder interactuar con el sistema.	1

#	Preguntas adicionales	Valoración
11	Pude hacer la tarea que deseaba fácilmente	5
12	Encontré la información que buscaba	5
13	Obtuve la ayuda requerida al usar el sistema	5
14	Recomendaría usar este sistema	5

Nombre: Francisco Javier Volpicella Lobo

Cargo: Profesional I&D Soporte

Fecha: 15/10/2012

Por favor, valore los siguientes aspectos del sistema utilizando una escala del 1 al 5, donde 1 es "**En completo desacuerdo**" y 5 es "**Completamente de acuerdo**". Su respuesta ayudará a mejorar la usabilidad del sistema.

#	Preguntas	Valoración
1	Creo que me gustaría utilizar este sistema con frecuencia.	5
2	Encontré el sistema innecesariamente complejo	5
3	Creo que el sistema es fácil de usar.	4
4	Creo que voy a necesitar el apoyo de un técnico para poder utilizar este sistema.	1
5	Me pareció que las diversas funciones en este sistema están bien integradas.	5
6	Creo que hay inconsistencia en este sistema.	1
7	Me parece que la mayoría de los usuarios aprenderá a utilizar este sistema muy rápidamente.	4
8	Me pareció que el sistema es complicado de usar.	1
9	Me sentí seguro de usar el sistema.	5
10	Me parece que necesito aprender muchas cosas antes de poder interactuar con el sistema.	2

#	Preguntas adicionales	Valoración
11	Pude hacer la tarea que deseaba fácilmente	5
12	Encontré la información que buscaba	4
13	Obtuve la ayuda requerida al usar el sistema	5
14	Recomendaría usar este sistema	5

Nombre: JUAN RAMOS

Cargo: Tutor de Pericia de Geomecánica y Estabilidad de Hoyos

Fecha: 15-10-2012

Por favor, valore los siguientes aspectos del sistema utilizando una escala del 1 al 5, donde 1 es "En completo desacuerdo" y 5 es "Completamente de acuerdo". Su respuesta ayudará a mejorar la usabilidad del sistema.

#	Preguntas	Valoración
1	Creo que me gustaría utilizar este sistema con frecuencia.	5
2	Encontré el sistema innecesariamente complejo	2
3	Creo que el sistema es fácil de usar.	5
4	Creo que voy a necesitar el apoyo de un técnico para poder utilizar este sistema.	1
5	Me pareció que las diversas funciones en este sistema están bien integradas.	4
6	Creo que hay inconsistencia en este sistema.	1
7	Me parece que la mayoría de los usuarios aprenderá a utilizar este sistema rápidamente.	5
8	Me pareció que el sistema es complicado de usar.	1
9	Me sentí seguro de usar el sistema.	5
10	Me parece que necesito aprender muchas cosas antes de poder interactuar con el sistema.	2

#	Preguntas adicionales	Valoración
11	Pude hacer la tarea que deseaba fácilmente	4
12	Encontré la información que buscaba	5
13	Obtuve la ayuda requerida al usar el sistema	5
14	Recomendaría usar este sistema	5

Nombre: Zamaira Chacón

Cargo: Analista TIC

Fecha: 16/10/2012

Por favor, valore los siguientes aspectos del sistema utilizando una escala del 1 al 5, donde 1 es "**En completo desacuerdo**" y 5 es "**Completamente de acuerdo**". Su respuesta ayudará a mejorar la usabilidad del sistema.

#	Preguntas	Valoración
1	Creo que me gustaría utilizar este sistema con frecuencia.	5
2	Encontré el sistema innecesariamente complejo	1
3	Creo que el sistema es fácil de usar.	4
4	Creo que voy a necesitar el apoyo de un técnico para poder utilizar este sistema.	1
5	Me pareció que las diversas funciones en este sistema están bien integradas.	4
6	Creo que hay inconsistencia en este sistema.	1
7	Me parece que la mayoría de los usuarios aprenderá a utilizar este sistema muy rápidamente.	4
8	Me pareció que el sistema es complicado de usar.	1
9	Me sentí seguro de usar el sistema.	5
10	Me parece que necesito aprender muchas cosas antes de poder interactuar con el sistema.	3

#	Preguntas adicionales	Valoración
11	Pude hacer la tarea que deseaba fácilmente	4
12	Encontré la información que buscaba	5
13	Obtuve la ayuda requerida al usar el sistema	5
14	Recomendaría usar este sistema	5

Anexo 5



Gerencia AIT
Automatización, Informática y Telecomunicaciones

Plan de Pruebas

< VenLog 2.0 >

Versión: < 1.0.0 >

< Octubre, 2012 >

Descripción del Documento

Es la colección formada por los casos de prueba y procedimientos de prueba. Este artefacto incluye el propósito de las pruebas, qué elemento se va a probar, las herramientas a utilizar y con qué recursos, así como el documento que va hacer entregado. Al tener el resultado de las pruebas se puede comparar lo obtenido con lo esperado.

En este artefacto también se reflejan las características de hardware y software que serán empleados para realizar el conjunto de las pruebas al sistema.

Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.0.0	30/08/12	Darwin Martínez	Documento de pruebas versión inicial

Índice de Contenido

1	Introducción	130
	• Alcance.....	130
	• Definiciones, Acrónimos y Abreviaturas.....	130
	• Documentos Relacionados.....	130
2	Antecedentes y Propósito	131
	2.1 Antecedentes.....	131
	2.2 Propósito de la Evaluación.....	131
	2.3 Motivadores de la prueba.....	131
3	Objetos a ser Evaluados	131
4	Ámbito de las Pruebas.....	131
	4.1 Dentro del Ámbito.....	131
	4.2 Fuera del Ámbito.....	131
5	Enfoque de las Pruebas	131
6	Herramientas para las Pruebas.....	134
	6.1 Software.....	134
	6.2 Hardware.....	134
	6.3 Configuraciones del ambiente de pruebas.....	134
7	Casos de Prueba.....	135
8	Prioridades	137
	8.1 Casos de Prueba por Características de Prioridad.....	137
	8.2 Flujos de Trabajo de Pruebas.....	138
9	Entregables.....	139
	9.1 Lista de Entregables de Pruebas.....	139
	9.2 Ficha: Escenarios por Caso de Uso.....	139
	9.3 Ficha: Resumen de Ciclos de Prueba.....	140

10	Criterio para el Inicio y Fin del Plan de Pruebas	141
10.1	Criterios de Inicio.....	141
10.2	Criterios de Fin.....	141
10.3	Criterios de Suspensión y Retomo de Actividades.....	141
11	Criterios para el Lanzamiento.....	141
11.1	Criterios de Evaluación.....	141
11.2	Clasificación de los errores.....	141
11.3	Resultados de la prueba.....	142
12	Riesgos	142
13	Reportes de Problemas y Resolución	142
14	Responsabilidades, Personal y Necesidades de Capacitación.....	144
14.1	Personal y Roles Necesarios.....	144

Plan de Pruebas

1 Introducción

- **Alcance**

Este plan de pruebas aplica para la integración y las pruebas de sistema para VenLog 2.0 en su modulo lector de registros. Este plan de pruebas aplica para todos los requerimientos definidos en el documento ERS.

- **Definiciones, Acrónimos y Abreviaturas**

Canal: es un objeto donde se almacena las muestras tomadas del pozo. Se presenta como una secuencia de datos indexados con respecto a la profundidad, el tiempo o alguna otra dimensión física del pozo. Estos datos vienen expresados como una matriz, de tamaño Mx1 en el caso de las curvas y MxN en caso de las imágenes.

Trama: es un objeto que define un tipo de estructura donde se enumera y se hace referencia a un conjunto de canales. En ellos se define el rango de profundidad y la dirección (ascendente o descendente) en que fue tomada la muestra.

IU: Interfaz de usuario.

ft: es una medida de longitud (pies).

CU: Caso de uso.

Pad: es el objeto con el que la herramienta de perfilaje resistiva toma sus muestras. Las herramientas de perfilaje resistiva pueden tener cuatro, seis u ocho pad.

- **Documentos Relacionados**

Título	Fecha	Organización	Identificador del documento
ERS VenLog 2.0	<dd/mm/aa>	PDVSA - Intevep	ERS 1
DAS VenLog 2.0 (Modulo lector de registros)	27/08/12	PDVSA - Intevep	DAS 1



2 Antecedentes y Propósito

2.1 Antecedentes

No se poseen antecedentes, ya que esta es la primera versión del plan de pruebas.

2.2 Propósito de la Evaluación

- Encontrar tantos errores como sea posible.
- Realizar pruebas de rendimiento y capacidad.
- Encontrar los problemas importantes y determinar los riesgos percibidos de la calidad.

2.3 Motivadores de la prueba

- Requerimientos Funcionales
- Requerimientos No Funcionales

3 Objetos a ser Evaluados

- Componente Lector DLIS
- Componente Generar Curva
- Componente Generar Imagen
- Componente Generar Imagen Resistiva

4 Ámbito de las Pruebas

4.1 Dentro del Ámbito

Dentro del ámbito del sistema se realizaran las siguientes pruebas:

- Pruebas de integración
- Pruebas del sistema
- Pruebas de aceptación

4.2 Fuera del Ámbito

Pruebas de portabilidad

5 Enfoque de las Pruebas

- T-01: Pruebas de integración

Objetivo:	Verificar la correcta interacción entre los componentes del sistema
Descripción:	Las pruebas de integración verifican si los componentes o subsistemas interactúan correctamente a través de sus interfaces, cubren la funcionalidad establecida, y se ajustan a los requerimientos especificados-
Técnicas:	<p><i>Pruebas de integración estructural:</i> estas pruebas son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirse a sentencias del lenguaje, se refieren a llamadas entre módulos</p> <p><i>Pruebas de integración funcional:</i> estas pruebas son similares a las pruebas de caja negra. Aquí se trata de encontrar fallos en la respuesta de un módulo cuando su operación depende de los procedimientos o funciones de otro(s) modulo(s).</p>
Fases:	Integrar la IU con el modulo lector y a partir de esta ir integrando los demás módulos.
Entorno de prueba:	Se realizarán pruebas que verifiquen la integración de todos los módulos del sistema, usando diferentes archivos de entrada.
Criterios de Éxito:	Los módulos funcionan correctamente de manera conjunta

• **T-02: Pruebas de Sistema**

Objetivo:	El objetivo principal de esta prueba es asegurar que el programa realice las funciones especificadas por el cliente en la fase de requerimientos.
Descripción:	En esta prueba se verifica si el sistema cumple con los objetivos planteados (Requisitos funcionales y no funcionales).
Técnicas:	Ejecutar cada CU, cada flujo de CU o función, usando data válida e inválida, para verificar: a) que los resultados ocurran cuando la data sea válida.; b) que se muestren apropiados mensajes de error o alerta cuando data inválida sea empleada.

Fases:	<ol style="list-style-type: none">1. Revisión lectura de archivo2. Revisión graficar curvas3. Revisión graficar imágenes3. Revisión graficar imagen resistiva4. Revisión exportar archivo5. Revisión cerrar archivo
Entorno de prueba:	Se realizará una prueba que verifique el correcto funcionamiento de todas las partes del sistema, usando diferentes archivos de entrada.
Criterios de Éxito:	El sistema funciona adecuadamente si se realizan todas las fases indicadas anteriormente y no ocurre ningún evento inesperado.

• **T-03: Pruebas de aceptación**

Objetivo:	El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad, usabilidad y rendimiento.
Descripción:	Esta prueba es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple los requerimientos establecidos.
Técnicas:	<i>Prueba alfa:</i> se lleva a cabo, por un cliente, en el lugar de desarrollo. El cliente usa el software de forma natural mientras el desarrollador lo observa y registra los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado. <i>Prueba beta:</i> se lleva a cabo por los usuarios finales del software en sus lugares de trabajo. A diferencia de la prueba alfa, el desarrollador no está presente normalmente, por lo tanto la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. Los usuarios registran todos los problemas que encuentren durante la prueba beta e informan a intervalos regulares al desarrollador.
Entorno de prueba:	Esta prueba se realizará con usuarios reales. La aplicación se instalará en los equipos de los usuarios para que ejecuten las pruebas.
Criterios de Éxito:	Si el usuario queda conforme con los resultados obtenidos.

6 Herramientas para las Pruebas

6.1 Software

Nombre	Versión	Descripción	Tipo de Prueba
Sistema Operativo	2002	Windows XP professional, service pack 3	Todas
VenLog	2.0	Aplicación propia, escrita en C++ con el Framework Qt 4.7	Todas
Qt Creator	2.2.1	IDE del Framework Qt	Sistema e integración
MinGW	3.81	Compilador de C++ basado en GCC para plataforma Win32	Sistema e integración

6.2 Hardware

Recurso	Cantidad	Descripción
Procesador	1	AMD Sempron (tm) 130 2,61GHz
RAM	1	896 mb
Disco duro	2	3. Capacidad 30GB y disponible 19GB 4. Capacidad 430GB y disponible 376GB

6.3 Configuraciones del ambiente de pruebas

Nombre de Configuración	Descripción	Implementación de la Configuración Física
Estándar	Configuración estándar para realizar las pruebas fuera del ambiente de desarrollo.	Instalador de la aplicación
Nativa	Configuración específica para pruebas dentro del ambiente de desarrollo	Compilador MinGw ID Qt Creator

7 Casos de Prueba

ID/Nombre/Sistema/Proyecto: VenLog 2.0		Nivel de Prueba: Parcial		
ID Caso de Uso:		Tipo(s) de Pruebas(s): de integración		
ID Requerimiento:		Ambiente de Prueba: Nativa		
ID/Nombre Escenario:		Autor del Caso de Prueba: Darwin Martínez		
ID/Nombre Caso de Prueba: CP-01		Nombre del Probador:		
Versión del Caso de Prueba: 1		Fecha de Creación: 31/05/12	Fecha de Ejecución:	
Condición(es) para que se ejecute el Caso de Prueba: Se deben cumplir los requerimientos de hardware, software y la configuración de ambiente de pruebas descritos en el ítem 6.				
Para la Ejecución del Caso de Prueba: Contar con las herramientas citadas anteriormente				
Elemento a probar	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Integración entre IU y modulo lector de registro.	Se debe poder acceder a las funciones del modulo lector de registro desde la IU.	100%	100%	100%
Integración entre IU, modulo lector de registro y modulo generar curva.	Se debe poder generar una(s) curva(s), por medio de la IU, a partir de los datos extraídos del archivo a través del lector.	100%	100%	100%
Integración entre IU, modulo lector de registro y modulo generar imagen	Se debe poder generar una(s) imagen(es), por medio de la IU, a partir de los datos extraídos del archivo a través del lector.	100%	100%	100%
Integración entre IU, modulo lector de registro y modulo generar imagen resistiva	Se debe poder generar una imagen resistiva, por medio de la IU, a partir de los datos extraídos del archivo a través del lector.	100%	100%	100%
Criterios de Aprobación del Caso de Prueba: Se debe cumplir con los resultados esperados en un 100%				
Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo: ____ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba: 31/05/12				
ID/Nombre/Sistema/Proyecto: VenLog 2.0		Nivel de Prueba: Parcial		
ID Caso de Uso:		Tipo(s) de Pruebas(s): de sistema		
ID Requerimiento:		Ambiente de Prueba: Nativa		

ID/Nombre Escenario:		Autor del Caso de Prueba: Darwin Martínez		
ID/Nombre Caso de Prueba: CP-02		Nombre del Probador:		
Versión del Caso de Prueba: 1		Fecha de Creación: 23/07/12	Fecha de Ejecución:	
Condición(es) para que se ejecute el Caso de Prueba: Se deben cumplir los requerimientos de hardware, software y la configuración de ambiente de pruebas descritos en el ítem 6.				
Para la Ejecución del Caso de Prueba: Contar con las herramientas citadas anteriormente				
Funcionalidad a probar	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Leer Archivo	En caso de ser un archivo valido mostrar los datos establecidos en los requerimientos, en caso contrario mostrar un mensaje del error ocurrido (archivo dañado, extensión distinta a .dlis, etc)	100%	100%	100%
Generar curva	Debe permitir construir una curva a través de los datos suministrados.	100%	100%	100%
Ver/Ocultar grilla	Debe permitir mostrar u ocultar una cuadrícula sobre el área de dibujo de la curva.	100%	100%	100%
Cambiar color	Debe permitir cambiar el color de la curva a través de un cuadro de dialogo.	100%	100%	100%
Generar imagen	Debe permitir construir una imagen relativa a la pared del pozo a través de los datos suministrados.	100%	100%	100%
Generar imagen resistiva	Debe permitir combinar los datos de la curva de alineación del norte magnético (azimut) y los datos de los pad de la herramienta resistiva para construir la imagen de las paredes del pozo.	100%	100%	100%
Cambiar paleta	Debe permitir cambiar la tonalidad de la imagen mediante las paletas de colores establecidas.	100%	100%	100%
Hacer zoom	Debe permitir aumentar o reducir el 25% de la escala vertical actual de la imagen o curva. También debe permitir restaurar la escala vertical, la imagen o curva, a su escala original.	100%	100%	100%
Criterios de Aprobación del Caso de Prueba: Se debe cumplir con los resultados esperados en un 100%				
Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo: ____ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba: 11/09/12				

ID/Nombre/Sistema/Proyecto: VenLog 2.0		Nivel de Prueba: Parcial		
ID Caso de Uso:		Tipo(s) de Pruebas(s): de aceptación		
ID Requerimiento:		Ambiente de Prueba: Estándar		
ID/Nombre Escenario:		Autor del Caso de Prueba: Darwin Martínez		
ID/Nombre Caso de Prueba: CP-03		Nombre del Probador:		
Versión del Caso de Prueba: 1		Fecha de Creación: 23/07/12	Fecha de Ejecución:	
Condición(es) para que se ejecute el Caso de Prueba: Se deben cumplir los requerimientos de hardware, software y la configuración de ambiente de pruebas descritos en el ítem 6.				
Para la Ejecución del Caso de Prueba: Contar con las herramientas citadas anteriormente				
Aspecto a probar	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Funcionalidad	Se debe cumplir con todas las funcionalidades definidas en la fase de requerimientos.	100%	100%	100%
Usabilidad	Los usuarios llenaran una encuesta denominada <i>System Usability Scale</i> (SUS). (Ver Anexo 1). Según Jeff Sauro, si el promedio del puntaje de las encuestas es mayor a 68% el sistema es usable (http://www.measuringusability.com/sus.php).	100%	68%	87,5%
Criterios de Aprobación del Caso de Prueba: Los resultados obtenidos deben ser mayor o igual a los resultados esperados.				
Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo: ____ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba:				

8 Prioridades

8.1 Casos de Prueba por Características de Prioridad

8.1.1 Esenciales

Pruebas de aceptación > funcionalidad > correctitud
 Pruebas de aceptación > usabilidad > comprensibilidad y legibilidad

8.1.2 Esperadas

- Pruebas de aceptación > funcionalidad > correctitud
- Pruebas de aceptación > funcionalidad > robustez
- Pruebas de aceptación > usabilidad > comprensibilidad y legibilidad
- Pruebas de aceptación > usabilidad > consistencia y familiaridad
- Pruebas de aceptación > usabilidad > satisfacción subjetiva

8.1.3 Deseadas

- Pruebas de aceptación > funcionalidad > correctitud
- Pruebas de aceptación > funcionalidad > robustez
- Pruebas de aceptación > funcionalidad > interoperabilidad
- Pruebas de aceptación > usabilidad > comprensibilidad y legibilidad
- Pruebas de aceptación > usabilidad > consistencia y familiaridad
- Pruebas de aceptación > usabilidad > satisfacción subjetiva
- Pruebas de aceptación > usabilidad > aprendizaje

8.2 Flujos de Trabajo de Pruebas

Flujo de trabajo para las pruebas	P2			P3			P4			P1		
	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar
1 Pruebas unitarias				Pruebas unitarias								
2 Pruebas lector				Pruebas lector								
3 Lector DLIS				Lector DLIS								
4 Nodo de datos				Nodo de datos								
5 Pruebas de interfaces				Pruebas de interfaces								
6 Enlace interfaz con lector DLIS				Enlace interfaz con lector DLIS								
7 Enlace interfaz con nodo de datos				Enlace interfaz con nodo de datos								
8 Prueba modulo generar curva				Prueba modulo generar curva								
9 Prueba modulo generar imagen				Prueba modulo generar imagen								
10 Prueba modulo generar imagen resistiva				Prueba modulo generar imagen resistiva								
11 Pruebas de integración				Pruebas de integración								
12 Integrar interfaz con modulo generar curva				Integrar interfaz con modulo generar curva								
13 Integrar interfaz con modulo generar imagen				Integrar interfaz con modulo generar imagen								
14 Integrar interfaz con modulo generar imagen resistiva				Integrar interfaz con modulo generar imagen resistiva								
15 Pruebas del sistema				Pruebas del sistema								
16 operaciones con archivo				operaciones con archivo								
17 operaciones con curvas				operaciones con curvas								
18 operaciones con imagen				operaciones con imagen								
19 operaciones con imagen resistiva				operaciones con imagen resistiva								
20 Pruebas de aceptación				Pruebas de aceptación								
21 Pruebas de funcionalidad				Pruebas de funcionalidad								
22 Pruebas de usabilidad				Pruebas de usabilidad								

9 Entregables

9.1 Lista de Entregables de Pruebas

Entregables	Descripción
Resumen de Ciclos de Prueba (RCP)	Este reporte contiene información: <ul style="list-style-type: none"> • Datos de la prueba ejecutada • Control de ejecución • Resultados de las pruebas

9.2 Ficha: Escenarios por Caso de Uso

ID Escenario	Flujo Básico	Flujo Alternativo 1
001 Leer Archivo	1. Indicar ruta 2. Abrir archivo 3. Verificar validez 4. Mostrar datos	1. Indicar ruta 2. Abrir archivo 3. Verificar validez 4. mostrar mensaje de error
002 Graficar Curva(s)	1. Leer Archivo 2. Seleccionar canales 3. Mostrar curva(s)	
003 Graficar Imagen(es) sónica(S)	1. Leer Archivo 2. Seleccionar canales 3. Mostrar imagen(es)	
004 Graficar Imagen Resistiva	1. Leer Archivo 2. Seleccionar canales 3. Verificar validez 4. Mostrar imagen	1. Leer Archivo 2. Seleccionar canales 3. Verificar validez 4. Mostrar mensaje de error
005 Cambiar color	1. Ejecutar CU 002 2. Escoger color 3. Mostrar curva con color seleccionado	
006 Ver/ocultar grilla	1. Ejecutar CU 002 2. Mostrar u ocultar grilla dependiendo del caso.	
007 Cambiar paleta	1. Ejecutar CU 003 2. Dar clic sobre la paleta hasta encontrar la tonalidad deseada.	

008 Hacer zoom	<ol style="list-style-type: none">1. Ejecutar CU 002, 003 o 0042. Aumentar o reducir la curva o imagen en un 25% de su tamaño actual.	<ol style="list-style-type: none">1. Ejecutar CU 002, 003 o 0042. Restaurar curva o imagen a su tamaño original.
----------------	--	---

9.3 Ficha: Resumen de Ciclos de Prueba

ID del Proyecto/ Nombre:

ID del Ciclo de Prueba:

Fechas para el Ciclo de Prueba: Desde: 31/08/12 Hasta: 10/10/12

ID Caso de Uso	ID Caso de Pruebas	Resultados Esperados	Resultados Obtenidos	Observación
001	1	95%		
002	1	95%		
003	1	95%		
004	1	95%		
005	1	100%		
006	1	100%		
007	1	100%		
008	1	100%		

Resultados/Observaciones para el Ciclo de Prueba:

El resultado que se obtuvo, estuvo a la altura de los resultados esperados, todo funcionó en base a lo que se tenía planeado en el diseño, cada una de las funcionalidades cumplieron la acción que debían realizar.

Aprobado Ciclo de Prueba por:

CLIENTE

ORGANIZACIÓN

PROBADOR

10 Criterio para el Inicio y Fin del Plan de Pruebas

10.1 Criterios de Inicio

- Contar con los equipos necesarios para poder llevar a cabo las pruebas
- Contar con el personal capacitado para realizar las pruebas

10.2 Criterios de Fin

Para que el plan de pruebas de VenLog 2.0 en su módulo lector de registros, se de por concluido, se deben cumplir las siguientes actividades.

- Realizar pruebas de integración.
- Probar el sistema
- Realizar pruebas de aceptación

Se realizarán las iteraciones necesarias hasta que la aplicación funcione conforme con lo establecido.

10.3 Criterios de Suspensión y Retomo de Actividades

La única manera de que las pruebas se suspendan es por la ausencia de alguna de las personas que realizarán las pruebas y estas se reanudarán en cuanto la persona se reincorpore o sea reemplazado por otra igualmente capacitada.

11 Criterios para el Lanzamiento

11.1 Criterios de Evaluación

Para que el plan de pruebas de VenLog 2.0 se dé por concluido se deben cumplir las siguientes actividades.

- Las pruebas del sistema deben haber arrojado un resultado favorable.
- El cliente debe haber probado y aceptado el programa.

11.2 Clasificación de los errores

Calificación	Definición de gravedad	Definición de prioridad
1	Fallo de segmentación	El error debe corregirse lo antes posible. El error bloquea el progreso en esta área.
2	Manejo y captura de excepciones	El error debe corregirse antes del lanzamiento del producto.

11.3 Resultados de la prueba

Se obtuvieron resultados positivos de todos los casos de prueba. No hubo errores sin resolver de Gravedad 1 y 2. Esto demuestra la consecución de los objetivos de las pruebas, con lo que la solución está lista para el lanzamiento.

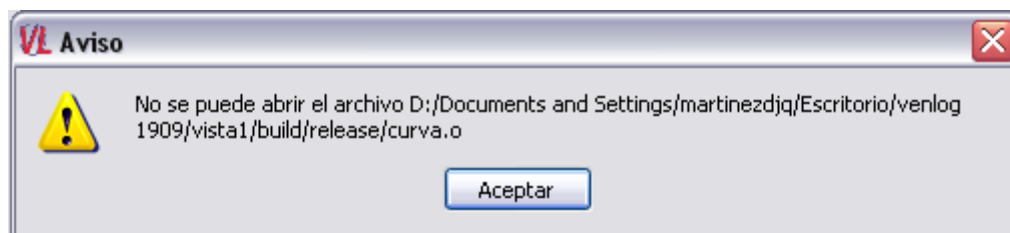
12 Riesgos

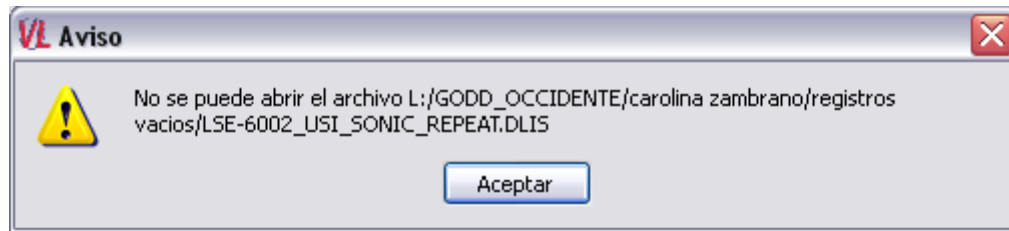
Riesgos	Plan de Contingencia	Impacto
Tiempo de pruebas mayor al previsto	<ul style="list-style-type: none">• Mejorar el plan de pruebas• Utilizar mayor número de personas en el plan de pruebas• Iniciar nuevamente con el plan de pruebas	Inconformidad por del parte del cliente.
Errores al ejecutar el plan de pruebas	<ul style="list-style-type: none">• Volver a hacer cada una de las pruebas e identificar los errores	En el área de pruebas y el área de desarrollo y mantenimiento de software

13 Reportes de Problemas y Resolución

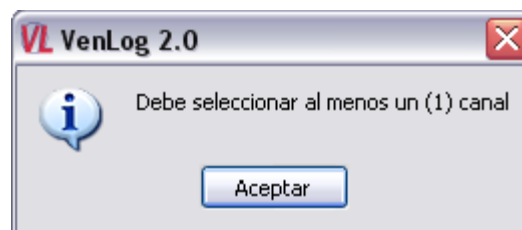
Los posibles errores que puede cometer el usuario durante la interacción con la aplicación son controlados por ventanas de aviso que aparecen cuando se va a ejecutar una acción que no puede ser procesada. Las ventanas de aviso que contempla VenLog 2.0 son las siguientes:

- *Intentar abrir un archivo dañado, vacío o de una extensión distinta a DLIS:*

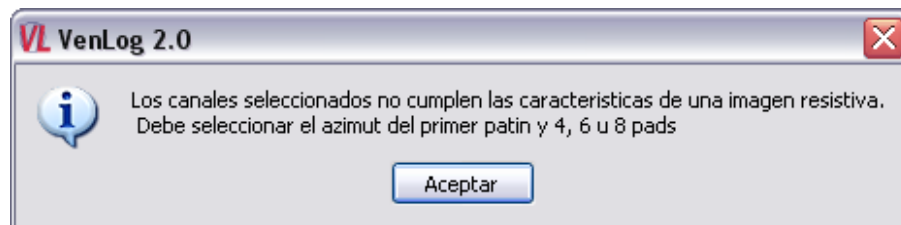




- *Intentar graficar datos con parámetros no validos:*
 - *Graficar sin seleccionar ningún check*



- *Graficar imagen resistivas con los parámetros incorrectos*



- *Graficar curvas superpuestas seleccionando al menos un canal de imagen.*



Existen otros errores reportados que no pueden ser controlados debido a que estos dependen del

rendimiento de la maquina en que se ejecute el programa. Estos errores se presentan cuando se grafican cierto número de canales y algunos de estos quedan fuera del alcance de la visión del usuario, por lo que es necesario mover la barra de desplazamiento horizontal hacia la derecha, cuando esto ocurre es posible que la curva o imagen presente en el área de dibujo, que estaba fuera del rango de visión del usuario, este incompleta, esto se soluciona moviendo la barra de desplazamiento vertical del área de dibujo que presenta la falla. Otro error similar ocurre cuando se desea ocultar la grilla, en ocasiones quedan fragmentos de esta en el área de dibujo, esto también se solventa moviendo la barra de desplazamiento vertical.

14 Responsabilidades, Personal y Necesidades de Capacitación

14.1 Personal y Roles Necesarios

Roles	Recursos Necesarios	Estado	Responsabilidades Específicas o Comentarios
Administrador del plan de pruebas	1	Asignado	• Coordinar que el plan de pruebas se lleve a cabo y hacer la planeación de este.
Usuarios de prueba	4	Pendiente	• Probar el sistema como si fueran los usuarios finales de la aplicación.

Copyright 2011 PDVSA – Gerencia Funcional AIT. Bajo licencia de Documentación Libre de GNU

Copyright 2009 PDVSA Intevep. Bajo licencia de Documentación Libre de GNU

Copyright (C) 2007 CNTI. Todos los derechos reservados.

Copyright (C) 2003 Jason Robbins. Todos los derechos reservados.

Copyright (C) 2006 CNTI. Todos los derechos reservados.