

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DESARROLLO DE UN SOFTWARE QUE PERMITA LA EDICIÓN Y
GRAFICACIÓN DE INFORMACIÓN RELACIONADA CON PERFILES DE POZO
MANEJADA EN EL DEPARTAMENTO DE EXPLORACIÓN Y PRODUCCIÓN DE LA
EMPRESA PDVSA PUERTO LA CRUZ”

Realizado por:
Martínez González, Julio Cesar

Trabajo especial de grado presentado en la Universidad de Oriente como requisito
parcial para optar al título de:

INGENIERO EN COMPUTACIÓN

Barcelona, Julio 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DESARROLLO DE UN SOFTWARE QUE PERMITA LA EDICIÓN Y
GRAFICACIÓN DE INFORMACIÓN RELACIONADA CON PERFILES DE POZO
MANEJADA EN EL DEPARTAMENTO DE EXPLORACIÓN Y PRODUCCIÓN DE LA
EMPRESA PDVSA PUERTO LA CRUZ”

Asesorado por:

Ing. Gabriela Veracierta
Asesor Académico

Barcelona, Julio 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DESARROLLO DE UN SOFTWARE QUE PERMITA LA EDICIÓN Y
GRAFICACIÓN DE INFORMACIÓN RELACIONADA CON PERFILES DE POZO
MANEJADA EN EL DEPARTAMENTO DE EXPLORACIÓN Y PRODUCCIÓN DE LA
EMPRESA PDVSA PUERTO LA CRUZ”

Jurado:

El Jurado hace constar que asignó a esta Tesis la calificación de:

EXCELENTE

Ing. Gabriela Veracierta

Asesor Académico

Ing. Claudio Cortínez

Jurado Principal

Ing. Víctor Mujica

Jurado Principal

Barcelona, Julio 2009

RESOLUCIÓN

De acuerdo al artículo 44 del reglamento de trabajos de grado:

“Los trabajos de grado son exclusiva propiedad de la universidad de oriente y sólo podran ser utilizados a otros fines con el consentimiento del consejo de nucleo respectivo, quien lo participará al consejo universitario”.

AGRADECIMIENTOS

A quien considero mi bendición, mi novia, mi esposa, Gladlyanna Jose, por su excelente e incomparable compañía, por siempre estar ahí, por haber sido mi punto de apoyo, inspiración y motivación durante mi carrera y el desarrollo de este proyecto.

A mis amigos Ruben Garcia, Ricardo Presilla, por su importante, valiosa e inigualable ayuda.

A todos los amigos y amigas que compartieron, colaboraron y me ayudaron en el transcurso de mi vida universitaria.

A la profesora Gabriela Veracierta por haberme asesorado y ayudado en este proyecto.

DEDICATORIA

Dedico este trabajo a Dios y la Virgen del Valle, a mis padres por su amor, su amistad, compañía y buenos consejos que me han ayudado en todo momento e incluso me han permitido mantener vivo el espíritu de lucha durante los tiempos más difíciles. A mi esposa que siempre estuvo a mi lado, dandome aliento cuando mas lo necesitaba y llenandome de fuerzas con sus palabras. A mi abuela, que con sus rezos me dio esperanza. A mi hermana, a mis sobrinas bellas, que todos los días me recuerdan la alegría de la vida y a toda mi familia, tíos y primos.

Dedico también este logro en honor y memoria de los familiares que ya no están físicamente pero siempre mantendré vivo más que su recuerdo: Al siempre grande abuelo Ildemaro, que desde donde este, se que vela por mi salud y bienestar y se que esta orgulloso de mi, mi abuelo Julio, mi abuela Blanca.

RESUMEN

Este proyecto tiene como finalidad la realización de un software que permita la edición y graficación de información relacionada con perfiles de pozo manejada en el Departamento de Exploración y Producción de la empresa PDVSA Puerto La Cruz. Con esto se plantea optimizar y reducir los tiempos y los errores, en la realización de las operaciones llevadas a cabo en la Gerencia. Este software permitirá unificar en solo un programa todas las aplicaciones que el departamento requiere: trabajar en software libre multiplataforma, fácil aplicabilidad en formatos PDF, JPG y TIF, así como la visualización y graficación de archivos LAS (Log ASCII Standard). Procedimientos todos que actualmente deben realizarse bajo sistemas operativos distintos, y por lo tanto en computadores diferentes. Los parámetros y requisitos serán obtenidos a través de los archivos y documentos recibidos por los diferentes departamentos involucrados en el proceso, a fin de desarrollar una herramienta que permita, generar resultados confiables, y reales, acorde con los datos obtenidos en físico, como los que obtienen en digital. La técnica a utilizar para el desarrollo de este proyecto seguirá los pasos del método científico, se estará utilizando la metodología Orientada a Objeto, basada en el ciclo de vida del Software y Lenguaje Unificado de Modelado (UML).

CONTENIDO

| | |
|--|-------------|
| RESOLUCIÓN | IV |
| AGRADECIMIENTOS..... | V |
| DEDICATORIA | VI |
| RESUMEN | VII |
| CONTENIDO..... | VIII |
| INDICE DE FIGURAS | XIII |
| INDICE DE TABLAS | XVII |
| CAPITULO I EL PROBLEMA | 18 |
| 1.1 PETRÓLEOS DE VENEZUELA S.A. | 18 |
| 1.2 PLANTEAMIENTO DEL PROBLEMA. | 20 |
| 1.3 OBJETIVOS. | 23 |
| <i>1.3.1 Objetivo General.</i> | 23 |
| <i>1.3.2 Objetivos Específicos.</i> | 24 |
| CAPITULO II MARCO TEÓRICO..... | 25 |
| 2.1 ANTECEDENTES | 25 |
| 2.2 MARCO METODOLÓGICO | 28 |
| 2.3 SOFTWARE | 29 |
| 2.4 TIPOS DE SISTEMAS DE INFORMACIÓN | 30 |
| 2.5 SOFTWARE LIBRE (EN INGLÉS FREE SOFTWARE) | 30 |
| 2.6 CARACTERÍSTICAS DEL SOFTWARE LIBRE | 31 |
| 2.7 UML (UNIFIED MODELING LANGUAGE)..... | 32 |
| 2.7.1 Modelos | 32 |
| 2.7.1.1 Modelo de Dominio..... | 32 |
| 2.7.2 Diagrama de UML..... | 33 |

| | |
|---|----|
| 2.7.3 Caso de uso..... | 33 |
| 2.7.4 Diagrama de Casos de Uso..... | 34 |
| 2.7.5 Actor | 34 |
| 2.7.6 Relaciones en un Diagrama de Casos de Uso..... | 35 |
| 2.7.7 Diagrama de Clases | 35 |
| 2.7.8 Diagrama de Objetos..... | 35 |
| 2.7.9 Diagramas de Secuencia | 35 |
| 2.7.9.1 Línea de Vida de un Objeto..... | 35 |
| 2.7.9.2 Activación..... | 36 |
| 2.7.9.3 Mensaje..... | 36 |
| 2.7.10 Diagrama de Colaboración..... | 36 |
| 2.7.10.1 Objeto | 36 |
| 2.7.10.2 Enlaces..... | 36 |
| 2.7.10.3 Flujo de mensajes | 37 |
| 2.7.10.4 Marcadores de Creación y Destrucción de Objetos..... | 37 |
| 2.7.10.5 Objeto compuesto | 37 |
| 2.7.10.6 Patrón de diseño..... | 37 |
| 2.7.10.7 Contexto..... | 38 |
| 2.7.11 Diagrama de Estados | 38 |
| 2.7.12 Diagrama de Actividades | 38 |
| 2.7.12.1 Componente..... | 38 |
| 2.7.13 Diagrama de Componentes | 39 |
| 2.7.14 Diagrama de Distribución..... | 39 |
| 2.7.15 Etapas y actividades en el desarrollo Orientado a objetos basado en UML | 39 |
| 2.7.16 Elementos Notacionales de UML | 40 |
| 2.8 LAS (LOG ASCII STANDARD)..... | 40 |
| 2.8.1 Estructura del encabezado del archivo LAS..... | 41 |
| 2.8.2 Limitación del formato LAS..... | 44 |

| | |
|--|-----------|
| 2.9 ARCHIVOS JPEG (.JPG) | 44 |
| 2.10 ARCHIVOS TIFF (.TIF) | 45 |
| 2.11 ARCHIVOS PDF (.PDF) | 46 |
| 2.11.1 Características de un archivo PDF | 46 |
| 2.12 REGISTROS DE POZOS | 48 |
| 2.12.1 Registros para la detección de la litología..... | 48 |
| 2.12.2 Registro para la detección de hidrocarburos..... | 48 |
| 2.12.3 Registro para la medición del diámetro del hoyo | 49 |
| 2.12.4 Registro para la medición de porosidad | 49 |
| 2.13 PROCESO DE INGENIERÍA DE SOFTWARE | 50 |
| 2.14 FASES DE LA INGENIERÍA DE SOFTWARE..... | 50 |
| 2.14.1 Análisis y Especificación de los Requerimientos | 50 |
| 2.14.2 Diseño..... | 51 |
| 2.14.3 Codificación..... | 51 |
| 2.14.4 Prueba del Sistema | 51 |
| 2.14.5 Instalación | 52 |
| 2.14.6 Mantenimiento del Sistema y Mejoras..... | 52 |
| 2.15 CICLO DE VIDA DEL SOFTWARE..... | 52 |
| 2.15.1 Ciclos de vida en cascada | 54 |
| 2.15.2 Descripción del Ciclo de vida en Cascada..... | 54 |
| | |
| CAPÍTULO III ANÁLISIS Y ESPECIFICACIÓN DE LOS | |
| REQUERIMIENTOS | 56 |
| 3.1 INTRODUCCIÓN | 56 |
| 3.2 ESTUDIO DEL CONTEXTO DEL SISTEMA | 57 |
| 3.2.1 Proceso de la GODD..... | 57 |
| 3.2.2 Roles y responsabilidades | 59 |
| 3.2.3 Modelo de Dominio | 60 |
| 3.2.4 Glosario de términos del Modelo de Dominio | 62 |

| | |
|--|-----------|
| 3.3 RIESGOS DEL SISTEMA | 62 |
| 3.3.1 Riesgos Críticos del Sistema..... | 62 |
| 3.4 REQUISITOS | 63 |
| 3.4.1 Requisitos Funcionales..... | 63 |
| 3.4.2 Requisitos no funcionales | 64 |
| 3.4.3 Requisitos de Software..... | 64 |
| 3.4.4 Requisitos de la Plataforma Hardware | 65 |
| 3.4.5 Identificación de los Actores | 65 |
| 3.5 CASO DE USO DEL SISTEMA | 66 |
| 3.5.1 Caso de uso Mostrar Registro Datos | 67 |
| 3.5.2 Caso de uso Mostrar Registro Imagen | 68 |
| 3.5.3 Caso de uso Graficar Curva..... | 69 |
| 3.5.4 Caso de uso Editar Archivo Datos | 71 |
| 3.6 ANÁLISIS | 73 |
| 3.6.1 Diagramas de clases de Análisis | 73 |
| 3.6.2 Identificación de Clases de Análisis | 74 |
| 3.6.3 Diagrama de clases de análisis Cargar Archivo de Datos | 76 |
| 3.6.4 Diagrama de clases de análisis archivo de Imagen..... | 77 |
| 3.6.5 Diagrama de clases de análisis Graficar Curva | 78 |
| 3.6.6 Diagrama de clases de análisis Editar Archivo Datos | 79 |
| CAPÍTULO IV DISEÑO | 81 |
| 4.1 FASE DE DISEÑO | 81 |
| 4.1.1 Diagrama de clases de colaboración para el caso de uso Graficar Curva..... | 81 |
| 4.2 DIAGRAMAS DE CLASES. | 83 |
| 4.3 DIAGRAMA DE SECUENCIA. | 83 |
| 4.4 DISEÑO DE INTERFACES. | 84 |
| 4.4.1 Interfaz Principal..... | 84 |

| | |
|---|------------|
| 4.4.2 Interfaz Abrir Archivo de Datos. | 85 |
| 4.4.3 Interfaz Graficar..... | 86 |
| CAPÍTULO V CONSTRUCCIÓN, CODIFICACIÓN Y PRUEBAS | 88 |
| 5.1 CONSTRUCCIÓN | 88 |
| 5.1.1 Interfaz Principal..... | 88 |
| 5.1.2 Interfaz Abrir Archivo de Datos. | 89 |
| 5.1.3 Interfaz Abrir Archivo de Imagen..... | 90 |
| 5.1.4 Interfaz Graficar..... | 91 |
| 5.2 CODIFICACIÓN | 97 |
| 5.2.1 Implementación del caso de uso Principal..... | 97 |
| 5.2.2 Implementación del caso de uso Abrir Archivo de Datos | 115 |
| 5.2.3 Implementación del caso de uso Graficar..... | 125 |
| 5.2.4 Implementación del caso de uso Editar Archivo de datos..... | 145 |
| 5.3 PRUEBAS..... | 167 |
| CONCLUSIONES | 169 |
| RECOMENDACIONES | 171 |
| BIBLIOGRAFÍA | 172 |
| APENDICE A MANUAL DE USUARIO | A-1 |
| A.1 INSTALACIÓN DEL PROGRAMA. | A-1 |
| A.2 EJECUTANDO EL PROGRAMA. | A-1 |
| A.3 ABRIENDO LA SESIÓN DEL USUARIO. | A-2 |

INDICE DE FIGURAS

| | |
|--|----|
| FIGURA 2.1 DIFERENTES SECCIONES DEL ENCABEZADO DE UN ARCHIVO LAS EN FORMATO ASCII..... | 43 |
| FIGURA 2.2 ETAPA GENÉRICA..... | 53 |
| FIGURA 2.3 CICLO DE VIDA EN CASCADA..... | 54 |
| FIGURA 3.1 FLUJO DE TRABAJO DE LA GODD. | 58 |
| FIGURA 3.2 MODELO DE DOMINIO..... | 61 |
| FIGURA 3.3 CASO DE USO PRINCIPAL DEL SISTEMA JGRAFEP. | 66 |
| FIGURA 3.4 CASO DE USO MOSTRAR REGISTRO DATOS DEL SISTEMA JGRAFEP..... | 67 |
| FIGURA 3.5 CASO DE USO MOSTRAR REGISTRO IMAGEN DEL SISTEMA JGRAFEP..... | 68 |
| FIGURA 3.6 CASO DE USO GRAFICAR CURVA DEL SISTEMA JGRAFEP.... | 70 |
| FIGURA 3.7 CASO DE USO EDITAR ARCHIVO DATOS DEL SISTEMA JGRAFEP..... | 72 |
| FIGURA 3.8 DIAGRAMA DE CLASES DE ANÁLISIS CARGAR ARCHIVO DE DATOS. | 77 |
| FIGURA 3.9 DIAGRAMA DE CLASES DE ANÁLISIS ARCHIVO DE IMAGEN. | 78 |
| FIGURA 3.10 DIAGRAMA DE CLASES DE ANÁLISIS GRAFICAR CURVA... | 79 |
| FIGURA 3.11 DIAGRAMA DE CLASES DE ANÁLISIS EDITAR ARCHIVO DATOS. | 80 |

| | |
|---|-----|
| FIGURA 4.1 DIAGRAMA DE CLASES PARA EL CASO DE USO GRAFICAR CURVA..... | 82 |
| FIGURA 4.2 DIAGRAMA DE SECUENCIA DEL PROCESO GRAFICAR. | 83 |
| FIGURA 4.3 BOSQUEJO GENERAL DE LA INTERFAZ PRINCIPAL. | 85 |
| FIGURA 4.4 BOSQUEJO GENERAL DE LAS INTERFACES DE ABRIR ARCHIVO DE DATOS..... | 86 |
| FIGURA 4.5 BOSQUEJO GENERAL DE LA INTERFAZ GRAFICAR..... | 87 |
| FIGURA 5.1 INTERFAZ PRINCIPAL DE JGRAFEP..... | 88 |
| FIGURA 5.2 FILECHOOSER DE LA OPCIÓN CARGAR ARCHIVO DE DATOS. | 89 |
| FIGURA 5.3 VENTANAS DE LA OPCIÓN CARGAR ARCHIVO DE DATOS. .. | 90 |
| FIGURA 5.4 FILECHOOSER DE LA OPCIÓN CARGAR ARCHIVO DE IMAGEN..... | 91 |
| FIGURA 5.5 FILECHOOSER DE LA OPCIÓN GRAFICAR..... | 92 |
| FIGURA 5.6 COMBOBOX ESCALA. | 94 |
| FIGURA 5.7 COMBOBOX TIPO DE GRAFICA..... | 94 |
| FIGURA 5.8 CELDA BOOLEANAS DE SELECCIÓN DE CURVAS. | 95 |
| FIGURA 5.9 COMBOBOX LADO DE LA GRAFICA. | 95 |
| FIGURA 5.10 COLORCHOOSER..... | 96 |
| FIGURA 5.11 RANGOS DE GRAFICACIÓN..... | 96 |
| FIGURA 5.12 VENTANA PRINCIPAL..... | 97 |
| FIGURA 5.13 VENTANAS CON LOS DATOS DEL ARCHIVO..... | 115 |
| FIGURA 5.14 VENTANA GRAFICAR. | 125 |

| | |
|---|------|
| FIGURA 5.15 VENTANA DE EDICIÓN DE ARCHIVO DE DATOS. | 145 |
| FIGURA A.1 VENTANA PRINCIPAL. | A-2 |
| FIGURA A.2 OPCIONES DEL MENÚ ARCHIVO..... | A-3 |
| FIGURA A.3 SELECTOR DE ARCHIVO DE DATOS. | A-4 |
| FIGURA A.4 VENTANAS CON LOS DATOS DEL ARCHIVO | A-5 |
| FIGURA A.5 SELECTOR DE ARCHIVO DE IMAGEN..... | A-5 |
| FIGURA A.6 SELECTOR DE ARCHIVO DE DATOS EN LA OPCIÓN GRAFICAR | A-6 |
| FIGURA A.7 SELECCIÓN DE ESCALA EN VENTANA DE GRAFICACIÓN.. | A-7 |
| FIGURA A.8 SELECCIÓN DE TIPO DE GRAFICA EN VENTANA DE GRAFICACIÓN | A-7 |
| FIGURA A.9 SELECCIÓN DE LAS CURVAS EN VENTANA DE GRAFICACIÓN | A-8 |
| FIGURA A.10 SELECCIÓN DE LA CASILLA EN LA CUAL SE COLOCARA EL LADO DE LA GRAFICA. | A-8 |
| FIGURA A.11 OPCIÓN LINEAL-LOGARÍTMICA. | A-9 |
| FIGURA A.12 OPCIÓN LINEAL-LINEAL..... | A-9 |
| FIGURA A.13 SELECCIÓN DE COLOR | A-10 |
| FIGURA A.14 INTRODUCCIÓN DE LOS VALORES DE LOS RANGOS..... | A-11 |
| FIGURA A.15 GRAFICA LISTA. | A-11 |
| FIGURA A.16 MENÚ DE EDICIÓN. | A-12 |
| FIGURA A.17 VENTANA DE EDICIÓN DE ARCHIVO DE DATOS..... | A-13 |
| FIGURA A.18 SELECTOR DE ARCHIVO EN VENTANA ORIGEN. | A-13 |

| | |
|--|------|
| FIGURA A.19 DATOS EN LA VENTANA ORIGEN. | A-14 |
| FIGURA A.20 SELECTOR DE ARCHIVO EN VENTANA FINAL..... | A-15 |
| FIGURA A.21 DATOS EN LA VENTANA FINAL..... | A-15 |
| FIGURA A.22 SELECCIÓN DE DATOS EN LA VENTANA ORIGEN. | A-16 |
| FIGURA A.23 COPIADO DE LA SELECCIÓN DE DATOS EN LA VENTANA ORIGEN..... | A-17 |
| FIGURA A.24 SELECCIÓN DE LA UBICACIÓN DE LOS DATOS EN LA VENTANA FINAL..... | A-18 |
| FIGURA A.25 PEGADO DE LA SELECCIÓN DE LOS DATOS EN LA VENTANA FINAL..... | A-19 |
| FIGURA A.26 CAMBIO DE NOMBRE DE LAS CURVAS EN LA VENTANA FINAL..... | A-20 |
| FIGURA A.27 CAMBIOS REALIZADOS EN LA VENTANA FINAL..... | A-21 |

INDICE DE TABLAS

| | |
|--|-----|
| TABLA 3.1 GLOSARIO DE TÉRMINOS DEL MODELO DE DOMINIO. | 62 |
| TABLA 3.2 TABLA DE IDENTIFICACIÓN DE ACTORES. | 65 |
| TABLA 3.3 TABLA DE IDENTIFICACIÓN DE CLASES DE CONTROL. | 74 |
| TABLA 3.4 TABLA DE IDENTIFICACIÓN DE CLASES DE INTERFAZ..... | 75 |
| TABLA 3.5 TABLA DE IDENTIFICACIÓN DE CLASES DE ENTIDAD. | 76 |
| TABLA 5.1 DESCRIPCIÓN DE LOS MENSAJES DE ERROR (1/2)..... | 167 |
| TABLA 5.1 DESCRIPCIÓN DE LOS MENSAJES DE ERROR (2/2)..... | 168 |

CAPITULO I

EL PROBLEMA

1.1 Petróleos de Venezuela S.A.

Petróleos de Venezuela S.A. es la corporación estatal de la República Bolivariana de Venezuela que se encarga de la exploración, producción, manufactura, transporte y mercadeo de los hidrocarburos, de manera eficiente, rentable, segura, transparente y comprometida con la protección ambiental; con el fin último de motorizar el desarrollo armónico del país, afianzar el uso soberano de los recursos, potenciar el desarrollo endógeno y propiciar una existencia digna y provechosa para el pueblo venezolano, propietario de la riqueza del subsuelo nacional y único dueño de esta empresa operadora.

Por mandato de la Constitución de la República Bolivariana de Venezuela, la totalidad de las acciones de Petróleos de Venezuela S.A. pertenecen al Estado Venezolano, en razón de la estrategia nacional y la soberanía económica y política, ejercida por el pueblo venezolano.

En ese sentido, PDVSA está subordinada al Estado Venezolano y por lo tanto actúa bajo los lineamientos trazados en los Planes de Desarrollo Nacional y de acuerdo a las políticas, directrices, planes y estrategias para el sector de los hidrocarburos, dictadas por el Ministerio de Energía y Petróleo.

La Corporación estatal, creada en 1975, por la Ley Orgánica que reserva al Estado la industria y el comercio de los hidrocarburos, cuenta con trabajadores comprometidos con la defensa de la soberanía energética y el deber de agregar el

mayor valor posible al recurso petrolero, guiados por los principios de unidad de comando, trabajo en equipo, colaboración espontánea y uso eficiente de los recursos.

La defensa de la Soberanía, es un valor transversal para todas las organizaciones de la Corporación y se enfoca en los conceptos de uso soberano de los recursos energéticos del país, e impulso a la soberanía tecnológica, esta última entendida como la creación e innovación de tecnologías de factura nacional, con vista hacia la generación de empleos de calidad, crecimiento económico y la creación de riqueza y bienestar para el pueblo venezolano.

PDVSA persigue la mayor efectividad en el ámbito de la gestión comunicacional, a través de la divulgación oportuna y efectiva de aquellos aspectos relevantes para el interés del pueblo venezolano, con el fin de facilitar el ejercicio efectivo de la contraloría social.

La transparencia y rendición de cuentas también constituyen un valor fundamental para PDVSA. En concordancia con este principio, la actuación del directorio, la alta gerencia y los trabajadores en general obedece a los mandatos de sobriedad, humildad, apego a los preceptos morales y administración sana y no ostentosa de los recursos propios y de la Corporación.

1.2 Planteamiento del problema.

En la empresa Petróleos De Venezuela S.A. actualmente cuenta con el software llamado GEOFRAME de la empresa Schlumberger con el que actualmente se hacen las labores de manipulación de la información de los perfiles de pozos, trabaja en la plataforma UNIX con un servidor de datos Oracle. Esta plataforma ha presentado dificultades por el crecimiento que ha tenido la empresa; debido a esto, ha surgido la necesidad de buscar la solución para el manejo y graficación de la información.

Entre los problemas que se encuentran en la empresa tenemos:

Una de las deficiencias de esta plataforma es que no se cuenta con la forma de visualizar archivos de imagen en formato PDF, por lo cual, cuando se va a comparar de manera visual los datos con las graficas con la imagen en dicho formato, es necesario abrir la imagen en un computador con un sistema operativo Microsoft Windows y hacer la comparación desde dos monitores pertenecientes a computadores distintos, lo cual hace dicha labor más difícil para el personal de la empresa.

Los usuarios del sistema constantemente se quejan, debido a lentitud con que la aplicación hace la apertura de los datos y las imágenes.

Los usuarios se encuentran inconformes debido a problemas con la red, ya que el servidor de aplicaciones pierde conexión con frecuencia, y esto impide a los usuarios ejecutar la aplicación.

La mayoría de los usuarios presentan desagrado a la interfaz grafica del software existente, ya que es poco amigable y robusta.

Las actividades que se desarrollan comúnmente con el software son:

- Apertura de archivos de datos LAS.
- Apertura de Archivos de Imagen JPG, TIFF, PDF.
- Visualización del Encabezado del archivo de datos.
- Edición de los datos correspondientes a los estudios hechos en los pozos (curvas).
- Selección de las Opciones de graficación.
- Graficación de las curvas.

En función de esta debilidad la empresa se propone el desarrollo de una aplicación basada en software libre que pueda trabajar en ambiente Windows y también en ambiente GNU-Linux, ya que la empresa está realizando un proceso de migración de sistema operativo por lo cual este proyecto será elaborado con herramientas de Software Libre rigiéndose por el Decreto Presidencial N° 3390, en gaceta N° 38095, de fecha 28 de diciembre de 2004, mediante el cual se dispone que “La Administración Pública Nacional deberá emplear prioritariamente Software Libre desarrollado con Estándares Abiertos, en todos sus Sistemas, Proyectos y Servicios Informáticos”.

El software a realizarse será una aplicación de escritorio sin necesidad de usar una base de datos, ya que estos son entregados en formato impreso y en CD’s, de manera que para hacer las comparaciones y las graficas se pueden realizar a través de dichos discos, de manera mas rápida y sin la utilización de la red para el cargado de los datos.

Este proyecto tendrá las siguientes limitaciones para ser llevado a buen término:

Físicas: La aplicación se desarrollará y se aplicará en el departamento de Exploración y Producción de planta baja y primer piso del Modulo A, contando con los equipos computacionales e instalaciones y equipos de red necesarios para la puesta en marcha de la aplicación.

Lógicas: A solicitud de la empresa la aplicación se codificará con el lenguaje de desarrollo de aplicaciones java 2.0.

Temporales: Para desarrollar la aplicación se cuenta con un periodo de seis (6) meses.

Este proyecto tiene la finalidad de desarrollar una herramienta de gestión de información que permita mejorar el manejo de los datos correspondientes a pozos petroleros en el departamento de exploración y producción de la empresa PDVSA sede Guaraguao, también debe realizar graficas de las curvas correspondientes a los estudios realizados en dichos pozos para permitirles a los usuarios poder hacer las comparaciones entre las curvas realizadas por el software y las curvas impresas que vienen desde los pozos. De esta forma se podrá simplificar e incrementar la productividad y eficiencia dentro de la empresa. Con esto se pretende formar una base en el desarrollo de otros proyectos basados en software libre para el uso de la empresa.

Este proyecto permitirá unificar en solo un programa todas las aplicaciones que la empresa requiere: trabajar en software libre multiplataforma, fácil aplicabilidad en formatos PDF, JPG y TIF, así como la visualización y graficación de archivos LAS. Procedimientos todos que actualmente deben realizarse bajo sistemas operativos distintos, y por lo tanto en computadores diferentes.

1.3 Objetivos.

1.3.1 Objetivo General.

Desarrollar un software que permita la edición y graficación de información relacionada con perfiles de pozo manejada en el departamento de exploración y producción, en la empresa PDVSA Puerto La Cruz

1.3.2 Objetivos Específicos.

- Analizar los requerimientos del software, mediante la recopilación de información de las operaciones en el departamento de exploración y producción de la empresa PDVSA con los datos de Pozos.
- Establecer los datos de entrada necesarios para la elaboración del software.
- Modelar los procesos del sistema, que satisfagan las necesidades de los usuarios, para el correcto desarrollo del mismo.
- Diseñar las interfaces de sistema.
- Realizar la codificación, depuración y documentación del software.

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes

Barreto, H. (2003) “Diseño De Un Sistema De Información De Inventario En El Departamento IT (Información Tecnología) de una Empresa Petrolera. Puerto La Cruz, Sector Venecia”. Trabajo de grado presentado en la Universidad de Oriente Núcleo de Anzoátegui para optar al título de Ingeniero de Sistemas, en el cual se rediseñó un sistema de inventario por razones logísticas y de estructura interna en la empresa. Gracias a la aplicación de UML se diseñó Sisvent, un Sistema de Inventario mejorado que toma en cuenta factores como la asignación de equipos a personal foráneo de la empresa, control de licencias para el software, generación de reportes automáticos por pantalla o impresora, también permite que un usuario pueda tener dos o más modelos iguales de un mismo equipo, en fin se diseñó de tal forma que la migración a la intranet queda servida.^[7]

London, M. (2004). “Desarrollo de un Software que permita el Monitoreo de la Información de Estado de un Portal Web Alojado en la Intranet Corporativa de PDVSA, utilizando la Nueva Plataforma de Microsoft.NET”. Trabajo de grado presentado en la Universidad de Oriente, Núcleo Anzoátegui para optar al título de Ingeniero en Computación. Este trabajo surgió debido a un requerimiento de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT), en Puerto la Cruz, Estado Anzoátegui, teniendo como fin el desarrollo de un Sistema que permita el monitoreo de la información de estado de un portal web alojado en la intranet corporativa de PDVSA, S.A. (SIMPOWEB). Para el desarrollo del proyecto se

empleó el Proceso Unificado de Desarrollo de Software tomando como base el Lenguaje Unificado de Modelado (UML).^[6]

Cardozo, D. (2007). “Desarrollo de un sistema que permita automatizar la información del historial de pozo, en la unidad de explotación de yacimientos de PDVSA-SANTOME”. Realizo un software de graficación e impresión para un sistema de control de torque computarizado de una empresa contratista de PDVSA. En este trabajo se utilizo el Lenguaje de Modelado Unificado para la representación de la arquitectura del software, se realizo utilizando la herramienta de desarrollo Power Builder 7.0 y para la gestión de base de datos se utilizo Oracle 9i.^[2]

Coa, L. (2007). “Desarrollo de un sistema que permita automatizar la información del historial de pozo, en la unidad de explotación del yacimiento de PDVSA – San Tome”. Para el desarrollo del proyecto fue utilizada la metodología del Proceso Unificado de Desarrollo de Software, para la codificación se utilizo Microsoft.Net Framework y el lenguaje orientado a objeto fue Visual Basic.Net.^[3]

Presilla, R. (2009). “Software de apoyo para la determinación y jerarquización de unidades de flujo, en función de índices de calidad de roca y saturación de fluidos producibles en yacimientos hidrocarburíferos, en la empresa INTER-ROCK. C.A.”, se plantea la necesidad de optimizar los procesos de jerarquización de unidades flujo, considerando los adelantos tecnológicos que se desarrollan día a día en la industria petrolera. Para dicha jerarquización se consideran parámetros como el radio de las gargantas porales, presión capilar, las permeabilidades relativas, a fin de determinar los volúmenes producibles de fluidos así como su movilidad a determinadas condiciones de saturación de agua. Dichos parámetros serán obtenidos a partir del PetroFlow System1.0, la codificación se realizó en JAVA NetBeans 5.5 y el manejo de base de datos se realizo a través de MySql Server 4.01.^[1]

2.2 Marco Metodológico

La metodología para implementar este proyecto y lograr un desarrollo óptimo y eficiente de la aplicación es: Metodología Orientada a Objetos de Martin y Odell, basándose en el ciclo de vida del Software y utilizando Lenguaje Unificado de Modelado (UML) mediante el desarrollo de casos de Uso, que permite representar los procesos llevados a cabo por el usuario en el sistema, donde se especifica que debería hacer el software, sin definir su implementación. Este modelo también permite definir los límites del sistema.

Esta metodología consta de las siguientes etapas:

Etapas I: Análisis y requerimientos del software

En esta etapa se realizará la recopilación de los requisitos del sistema para su posterior análisis. Para proceder al levantamiento de información, se realizarán breves entrevistas con los usuarios involucrados en el desarrollo del proceso de estudio de curvas de pozos, y de esta manera lograr que el software sea lo más apegado posible a los requerimientos de los usuarios finales.

Duración estimada: 4 semanas.

Etapas II: Diseño del software

En esta etapa se evaluará toda la información recopilada para proceder a diseñar la aplicación; apoyándose en la Metodología Orientada a Objetos basándose en el ciclo de vida del Software y utilizando el Lenguaje Unificado de Modelado (UML) mediante el desarrollo de los casos de Uso del proceso actual en la empresa, lo cual permitirá modelar interfaces sencillas, funcionales y eficientes.

Duración estimada: 2 semanas.

Etapa III: Codificación, compilación y documentación del software

En esta etapa se llevará a cabo la concepción de la herramienta de software; partiendo de la información recopilada de los usuarios en etapas previas; una vez diseñado el software se procederá a codificarlo y compilarlo. Además en esta etapa se elaboran también la documentación correspondiente al software.

Duración estimada: 14 semanas.

Etapa IV: Pruebas

Durante esta etapa se procederá a realizar las pruebas pertinentes al software a manera de comprobar su eficiencia y veracidad en la emisión de resultados.

Duración estimada: 8 semanas.

2.3 Software

Es un conjunto de instrucciones que se encarga de dirigir el hardware del computador para realizar una o varias acciones y también se encarga de tomar decisiones lógicas. Al agrupar una colección o combinación de programas, procedimientos, datos y equipamiento utilizado en el procesamiento de información, estamos en presencia de un Sistema, los cuales pueden clasificarse como sigue:

Sistema Operativo: “Es el programa del sistema que controla todos los recursos del computador y ofrece el soporte básico sobre el cual pueden escribirse los programas de aplicación.”^[8]

Sistema de información: “Es una organización el conjunto total de procedimientos, operaciones y funciones dedicadas a la generación y difusión de datos e información.”^[11]

2.4 Tipos de Sistemas de Información

Transaccionales: Este tipo de sistemas de información se encargan de manejar operaciones ordinarias como lo son: la facturación de compras ó ventas, entre otras.

Sistemas de apoyo de las decisiones:

ASS: Sistema para el apoyo a las decisiones.

GDSS: Sistema para el apoyo de las decisiones grupales.

EIS: Sistemas de información para ejecutivos.

EDSS: Sistema para el apoyo a las decisiones utilizando sistemas expertos.

Sistemas Estratégicos: Son aquellos sistemas que ofrecen ventajas competitivas, poseen una vida útil corta.

Sistemas Administrativos Integrales: Son programas que se encargan de gestionar ó coordinar las operaciones administrativas de un departamento ó área específica.^[11]

2.5 Software libre (en inglés free software)

Es la denominación del software que respeta la libertad de los usuarios y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a cuatro libertades de los usuarios del software: La libertad de usar el programa, con cualquier propósito, de estudiar el funcionamiento del programa, adaptarlo a las necesidades, de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo).

El software libre suele estar disponible gratuitamente, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo que no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial").

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquél que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público. ^[11]

2.6 Características del Software Libre

FLEXIBILIDAD. Si el código fuente está disponible, los desarrolladores pueden aprender y modificar los programas a su antojo, adaptándolo para realizar tareas específicas. Además, se produce un flujo constante de ideas que mejora la calidad de los programas.

FIABILIDAD Y SEGURIDAD. Con varios programadores a la vez mirando el mismo trabajo, los errores se detectan y corrigen antes, por lo que el producto resultante es más fiable y eficaz que el comercial.

RAPIDEZ DE DESARROLLO. Las actualizaciones y ajustes se realizan a través de una comunicación constante vía Internet. Menores tiempos de desarrollo debido a la amplia disponibilidad de herramientas y librerías.

RELACIÓN CON EL USUARIO. El programador se acerca mucho más a la(s) necesidad(es) real(es) de su cliente, y puede crear un producto específico para él.

LIBRE. Es de libre distribución, cualquier persona puede regalarlo, venderlo o prestarlo.

COMBATE EFECTIVAMENTE LA PIRATERÍA DE SOFTWARE. ^[11]

2.7 UML (Unified Modeling Language)

“Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.” ^[5]

2.7.1 Modelos

Son representaciones abstractas de un programa considerando un propósito específico. ^[5]

2.7.1.1 Modelo de Dominio

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis.

Con el modelo del dominio se capturan los objetos más importantes en el contexto del Software

Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno del sistema. El objetivo de este modelado es comprender el contexto y los requerimientos del sistema propuesto; en otras palabras el modelado del dominio contribuye a entender el problema, que se supone que el sistema resuelve en relación a su contexto.

Los objetos del dominio se relacionan a través de asociaciones, agregaciones y composiciones para modelar el desenvolvimiento de las actividades; ayudando a definir “qué” deberá hacer el sistema para resolver el problema y no “cómo” lo hará.

2.7.2 Diagrama de UML

Es una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos. ^[5]

2.7.3 Caso de uso

Se representa en el diagrama por una elipse, denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores (usuarios del sistema) y por el sistema en un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema. Va acompañado de un nombre significativo. Los casos de usos pueden relacionarse entre sí a través de asociaciones que permiten, entre otras cosas, refinar el Modelo de Casos de Usos a través de las asociaciones de:

Inclusión (asociación estereotipada como <<include>>). Permite incorporar el flujo de eventos de un caso de uso pequeño dentro de un caso de uso base de la aplicación.

Extensión (asociación estereotipada como <<extend>>). Permite incorporar el flujo de eventos de un caso de uso pequeño dentro de un caso de uso base de la aplicación bajo la ocurrencia de una determinada condición, cuando la misma evalúa verdadero.

Generalización (asociación estereotipada como <<generalization>>). Permite establecer una jerarquía de herencia al nivel de los casos de uso, donde el caso de uso derivado adquiere toda la especificación del caso de uso base e incorporar nuevos requerimientos a la especificación.^[5]

2.7.4 Diagrama de Casos de Uso

Es una técnica para capturar información respecto de los servicios que un sistema proporciona a su entorno. Los diagramas de Casos de Uso muestran las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).^[5]

2.7.5 Actor

Es un usuario del sistema, que necesita o usa algunos de los casos de uso. Se representa mediante un , acompañado de un nombre significativo, si es necesario.^[5]

2.7.6 Relaciones en un Diagrama de Casos de Uso

Entre los elementos de un diagrama de Casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas entre ellos (del elemento dependiente al independiente).^[5]

2.7.7 Diagrama de Clases

Es una representación de las clases del sistema con sus relaciones estructurales y de herencia.^[5]

2.7.8 Diagrama de Objetos

Un objeto es una instancia de clase (una entidad que tiene valores específicos de los atributos y acciones).^[5]

2.7.9 Diagramas de Secuencia

Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación.^[5]

2.7.9.1 Línea de Vida de un Objeto

Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos (véase 2.7.9.2). El rectángulo de encabezado contiene el nombre del objeto y el de su clase, en un formato *nombreObjeto: nombreClase*.^[5]

2.7.9.2 Activación

Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto. ^[5]

2.7.9.3 Mensaje

El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta. ^[5]

2.7.10 Diagrama de Colaboración

Modela la interacción entre los objetos de un Caso de Uso. Ofrece una mejor visión del escenario cuando el analista está intentando comprender la participación de un objeto en el sistema. A diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales,...) y ciclos en la ejecución. ^[5]

2.7.10.1 Objeto

Un objeto se representa con un rectángulo, que contiene el nombre y la clase del objeto en un formato *nombreObjeto: nombreClase*. ^[5]

2.7.10.2 Enlaces

Son instancias de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos. Esta acompañada por un número que indica

el orden dentro de la interacción y por un estereotipo que indica que tipo de objeto recibe el mensaje. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Los estereotipos indican si el objeto que recibe el mensaje es un atributo (association y se asume por defecto), un parámetro de un mensaje anterior, si es un objeto local o global. ^[5]

2.7.10.3 Flujo de mensajes

Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace. ^[5]

2.7.10.4 Marcadores de Creación y Destrucción de Objetos

Puede mostrarse en la gráfica cuáles objetos son creados y destruidos, agregando una restricción con la palabra *new* o *delete*, respectivamente, cercana al rectángulo del objeto. ^[5]

2.7.10.5 Objeto compuesto

Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. ^[5]

2.7.10.6 Patrón de diseño

Un diagrama de colaboración puede especificar un contrato entre objetos, parte esencial para la descripción de un patrón de diseño. Este diagrama contiene todos los elementos citados de un diagrama de colaboración, dejando libres posiblemente los

tipos exactos de algunos objetos o con nombres genéricos para los mensajes. Una "instancia" del patrón se representa como una elipse unida mediante flechas puenteadas a los objetos o clases que participan realmente en el patrón. Estas flechas pueden tener roles, indicando cuál es el papel de cada elemento dentro del patrón. ^[5]

2.7.10.7 Contexto

Un contexto es una vista de uno o más elementos dentro del modelo que colaboran en el desarrollo de una acción. Se usa para separar los demás elementos en el modelo de este problema en particular y darle énfasis. Puede mostrar solo los detalles relevantes de las clases u objetos que contiene, para resaltar su utilidad. ^[5]

2.7.11 Diagrama de Estados

Son modelos del comportamiento de una parte del sistema. ^[5]

2.7.12 Diagrama de Actividades

Se utiliza para especificar el comportamiento de los objetos de una clase, la lógica de una operación o método, una parte o toda la descripción de un Caso de uso y la descripción de un Flujo de Trabajo. ^[5]

2.7.12.1 Componente

Es una colección de objetos o componentes más pequeños que interaccionan entre ellos y se combinan para dar un servicio. Es similar a una caja negra, en la cual los servicios del componente se especifican por su interfaz o interfaces, sin ofrecer conocimiento del diseño e implementación internas del componente. Los

componentes se representan en el diagrama de clases de UML especificando la interfaz de una clase o paquete. Hay dos notaciones para mostrar una interfaz - una es mostrar la interfaz como una 'regular class symbol' con el estereotipo "interfz", con una lista de operaciones soportadas por esta interfaz, detalladas en el 'operation department' (departamento de operación). 'The alternate, shortcut notation' es mostrar la interfaz como un círculo pequeño junto con la clase con una línea sólida, con el nombre de la interfaz en el círculo. ^[5]

2.7.13 Diagrama de Componentes

Es representación de la estructura del software y la dependencia entre sus componentes. ^[5]

2.7.14 Diagrama de Distribución

Es la representación de la distribución en tiempo de ejecución de los elementos de procesamiento y componentes de software, junto a los procesos y objetos asociados al mismo. ^[5]

2.7.15 Etapas y actividades en el desarrollo Orientado a objetos basado en UML

En la versión definitiva de la metodología publicada en 1999 por Booch, Rumbaugh y Jacobson que se basa en las fases para el modelado de un sistema de información, se pueden tener en cuenta las siguientes etapas:

Análisis de Requerimientos.

Diseño del sistema.

Diseño detallado.

Implementación y pruebas.

Cada etapa consta de actividades que le dan cuerpo y los documentos que se esperan al final de cada una de ellas.

2.7.16 Elementos Notacionales de UML

Los elementos notacionales que presenta el UML pretenden ser un lenguaje común para el modelamiento de cualquier sistema y se agrupan en tipos de diagrama:

Diagrama de Clases.

Diagrama de Modelo de Dominio.

Diagrama de Objetos.

Diagrama de Casos de Uso.

Diagrama de Secuencia.

Diagrama de Colaboración.

2.8 LAS (Log ASCII Standard)

La Sociedad Canadiense de Taladro de Pozos, Comité de disquetes, ha diseñado un formato estándar para registrar datos en disquetes. Es conocido como el formato LAS siglas en inglés (Log ASCII Standard).

El archivo LAS contiene datos en formato ASCII con una mínima información en el encabezado y se destina a presentar ópticamente los registros de las curvas.

El objetivo del formato LAS es suplir la información básica de registro de datos digitales a los usuarios de computadoras personales en un formato que sea rápido y fácil de usar.

Las siguientes versiones de archivos LAS tal como define la Sociedad Canadiense de Taladro de Pozos son aceptadas como entrada:

- LAS Version 1.0
- LAS Version 2.0

Aunque no hay límite especificado en la norma de Archivos LAS para la longitud del mnemotécnico, la mayoría del software sólo acepta 64 caracteres en este campo.

Para ser utilizados por las aplicaciones, todos los archivos LAS deberían seguir el formato aprobado de las palabras mnemotécnicas estándar. Cualquier variación de la norma de palabras mnemotécnicas errara la exploración del archivo. ^[4]

2.8.1 Estructura del encabezado del archivo LAS

El encabezado del archivo LAS esta compuesto de las siguientes secciones principales:

Version Information

Esta sección describe la versión del formato de la cinta, el indicador de ajuste o no ajuste, y, si es aplicable, una descripción de los parámetros de ajuste (por ejemplo “one line per depth Step”). ^[4]

Well Information.

Esta sección contiene información permanente que identifica el pozo, como el nombre del pozo, nombre del campo, localización, valores de profundidad y así sucesivamente. Esta sección es obligatoria.^[4]

Curve Information

Esta sección describe las curvas y sus unidades en el orden en que aparecen en la sección de datos del archivo. Esta sección es obligatoria.^[4]

Parameter Information

Esta sección describe los valores de entrada de varios parámetros que pertenecen al pozo. Esta sección es opcional.^[4]

Other

Esta sección contiene las observaciones opcionales o comentarios.^[4]

Data

Esta es la sección del registro de datos ASCII. En formato ASCII, las líneas pueden estar en un modo de ajuste o el modo de no ajuste.^[4]

La siguiente figura muestra las diferentes secciones del encabezado de un archivo LAS en formato ASCII.

El tilde ~ indica el comienzo de una nueva sección.

```

~VERSION INFORMATION BLOCK
VERS.          2.0:  CWLS Log Ascii Standard - Version 2.0
WRAP.          NO:   One Line per depth step

~Well Information Block
#MNMN.UNIT     DATA          DESCRIPTION
#-----
STRT.F         530.0000          : START DEPTH
STOP.F         15895.0010         : STOP DEPTH
STEP.F         0.5000           : STEP LENGTH
NULL.          -999.2500         : NULL VALUE
COMP.          SHELL OIL CO      : COMPANY
WELL.          OCSG 0708         NO 2 : WELL
FLD.           GA BLK 287        : FIELD
LOC.           427060009300      : LOCATION
CTRY.          926              : COUNTRY
SRVC.          SCHLUMBERGER      : SERVICE COMPANY
DATE.          10-MAR-66        : LOG DATE
UWI.           427060009300      : UNIQUE WELL ID
SON.           :                 : SERVICE ORDER NO.
LIC.           :                 : LICENSE NO.

~Curve Information Block
#MNMN.UNIT     API CODE      Curve Description
#-----
DEPT.F         00 960 00 00:    1 Depth
SP.MV          05 010 01 01:    2 SELF POTENTIAL
ASN.OHMM       05 041 16 01:    3 AMPLIFIED SHORT NORMAL
CILD.MMHO      05 110 46 01:    4 INDUCTION CONDUCTIVITY
ILD.OHMM       05 120 46 01:    5 INDUCTION RESISTIVITY

~Parameter Information Block
#MNMN.UNIT     DATA          DESCRIPTION
#-----
RW.            :                 : RW
RWT.DEGF      :                 : RW TEMPERATURE
EKB.F         :                 : ELEVATION KELLY BUSHING
#ERCB required information-----
RUN.          1              : RUN NO.
EGL.F         :                 : ELEVATION OF GROUND LEVEL
DREF.F        :                 : DEPTH REF
EREF.F        :                 : ELEVATION OF DEPTH REFERENCE
TDL.F         3032.0000      : TOTAL DEPTH, LOGGER
TDD.F         3032.0000      : TOTAL DEPTH, DRILLER
BOTL.F        3031.0000      : BOTTOM OF LOGGED INTERVAL
TOPL.F        529.0000       : TOP OF LOGGED INTERVAL
CSGL.F        529.0000       : CASING BOTTOM, LOGGER
CSGD.F        530.0000       : CASING BOTTOM, DRILLER
CSGS.IN       20.0000        : CASING SIZE
CSGW.         :                 : CASING WEIGHT
BS.IN         17.5000        : BIT SIZE

~Other
The original data was obtained from field tapes.

~A  Depth      SP          ASN          CILD          ILD
530.0000      2.8027      0.5260      982.9688      1.5016
530.5000      -4.6611     0.6282     952.4531     1.4617
531.0000      -8.6638     0.6796     921.9375     1.4218
531.5000     -21.1733     0.7312     888.4492     1.3840
532.0000     -25.9685     0.7832     851.0547     1.3619
532.5000     -28.2585     0.8270     810.4844     1.3397

```

~A Indica el comienzo del registro de datos ASCII

Figura 2.1 Diferentes secciones del encabezado de un archivo LAS en formato ASCII. Fuente Queipo, N.

2.8.2 Limitación del formato LAS

En el formato LAS, la unidad del mnemotécnico se escribe de la siguiente manera:

MNEM.UNIT

Donde *MNEM* es el mnemotécnico para el parámetro — por ejemplo, ASN — y *UNIT* indica el tipo de medida — por ejemplo, OHMM. Un punto (.) Separa el mnemotécnico de la unidad. Ningún otro punto se permite en la representación de la unidad. Por lo tanto, OHMM no puede ser escrito como OHM.M porque el punto Se utiliza para separar ASN de OHMM: ASN.OHMM.^[4]

Mnemotécnicos de curvas en archivos LAS deben contener cuatro caracteres. Si el mnemotécnico tiene menos de cuatro caracteres, el o los caracteres faltantes se sustituyen por un espacio o varios espacios sea el caso. Por ejemplo, el mnemotécnico ASN está escrito en el formato LAS como ASN[espacio].OHMM, de modo que en el encabezado de la cinta que se vería así:^[4]

ASN .OHMM

El Mnemotécnico SP se escribirá de la siguiente manera:

SP .MV

2.9 Archivos JPEG (.jpg)

El formato JPEG (Joint Photographic Experts Group, Grupo de expertos fotográficos unidos) se suele usar para ver fotografías y otras imágenes de tono continuo en archivos HTML en Internet y otros soportes en línea. El formato JPEG admite los modos de color CMYK, RGB y escala de grises. A diferencia de GIF, JPEG conserva toda la información del color de una imagen RGB.^[11]

JPEG utiliza un esquema de compresión ajustable con pérdidas que reduce de forma eficaz el tamaño del archivo identificando y descartando los datos no imprescindibles para la visualización de la imagen. Un nivel mayor de compresión genera una calidad de imagen inferior; un nivel inferior de compresión genera una mejor calidad de imagen, pero un tamaño de archivo mayor. En la mayoría de los casos, no se perciben diferencias entre la imagen resultante y la original cuando ésta se comprime con la opción de calidad Máxima. Al abrir una imagen JPEG, ésta se descomprime de forma automática.^[11]

Nota: la codificación JPEG, que puede efectuarse en un archivo EPS o DCS en una aplicación de edición de imágenes como Photoshop, no crea un archivo JPEG. En cambio, lo que hace es comprimir el archivo, gracias al esquema de compresión de JPEG descrito anteriormente.

JPEG funciona bien con fotografías, pero las imágenes JPEG de color sólido (imágenes que contienen grandes extensiones de un color) tienden a perder nitidez. InDesign reconoce y admite los trazados de recorte en los archivos JPEG creados en Photoshop. JPEG puede utilizarse tanto para documentos en línea como para documentos de impresión comercial; consulte a su proveedor de servicios de preimpresión para conservar la calidad de JPEG en la impresión.^[11]

2.10 Archivos TIFF (.tif)

TIFF es un formato flexible de imágenes de mapa de bits que prácticamente admiten todas las aplicaciones de pintura, edición de imágenes y maquetación de páginas. Asimismo, prácticamente todos los escáneres de escritorio pueden producir imágenes TIFF.^[11]

El formato TIFF admite archivos CMYK, RGB, de escala de grises, Lab, de color indexado y de mapas de bits con canales alfa y tintas planas. Puede seleccionar un canal de tinta plana al colocar un archivo TIFF. Los canales de tintas planas aparecen en InDesign como tintas planas en el panel Muestras.

Puede usar un programa de edición de imágenes como Photoshop para crear un trazado de recorte si desea incorporar un fondo transparente para una imagen TIFF. InDesign admite trazados de recorte en imágenes TIFF y reconoce los comentarios OPI codificados.

2.11 Archivos PDF (.pdf)

PDF (acrónimo del inglés *Portable Document Format*, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto). Está especialmente ideado para documentos susceptibles de ser impresos, ya que especifica toda la información necesaria para la presentación final del documento, determinando todos los detalles de cómo va a quedar, no requiriéndose procesos anteriores de ajuste ni de maquetación. Cada vez se utiliza más también como especificación de visualización, gracias a la gran calidad de las fuentes utilizadas y a las facilidades que ofrece para el manejo del documento, como búsquedas, hiperenlaces, etc.^[11]

2.11.1 Características de un archivo PDF

Es multiplataforma, es decir, puede ser presentado por los principales sistemas operativos (Windows, Linux o Mac), sin que se modifiquen ni el aspecto ni la estructura del documento original.^[11]

Puede integrar cualquier combinación de texto, gráficos, imágenes e incluso música.

Es uno de los formatos más extendidos en Internet para el intercambio de documentos. Por ello es muy utilizado por empresas, gobiernos e instituciones educativas.

Es una especificación abierta, para la que se han generado herramientas de Software Libre que permiten crear, visualizar o modificar documentos en formato PDF. Un ejemplo es la suite ofimática OpenOffice.org y el procesador de textos LaTeX.

Puede cifrarse para proteger su contenido e incluso firmarlo digitalmente.

El archivo PDF puede crearse desde varias aplicaciones exportando el archivo, como es el caso de los programas de OpenOffice.org y también podrá hacerlo con el Service Pack 2 (SP2) del paquete ofimático Microsoft Office 2007, con lanzamiento anunciado por Microsoft para algún momento del primer semestre de 2009.

Es el estándar ISO (ISO 19005-1:2005) para ficheros contenedores de documentos electrónicos con vistas a su preservación de larga duración

El formato PDF, es ahora un estándar formal abierto conocido como ISO 32000. Mantenido por la Organización Internacional de Normalización (ISO), la norma ISO 32000 seguirá desarrollándose para cumplir los objetivos de proteger la integridad y longevidad del formato PDF, lo que proporciona un estándar abierto para los más de mil millones de archivos PDF que existen hoy en día.

2.12 Registros de Pozos

Los Registros son representaciones de medidas continuas de señales a lo largo del pozo, las cuales son procesadas para identificar litología, posible presencia de hidrocarburos y porosidad, entre otros. ^[4]

2.12.1 Registros para la detección de la litología

Registro de potencial espontáneo o curva SP. Se basa en la medida de la diferencia del potencial eléctrico natural existente en la formación a diferentes profundidades. La medición del SP requiere de un electrodo fijo en la superficie y otro móvil inmerso en el pozo. ^[4]

Curva de Rayos Gamma. La curva de rayos gamma se basa en la medición de la radiación natural de la formación la cual refléjale contenido de arcilla o lutita de las rocas sedimentarias. ^[4]

2.12.2 Registro para la detección de hidrocarburos

Registro de Resistividad. Miden la resistividad de la formación a diferentes profundidades. La resistividad, se define como la capacidad que tiene un material o sustancia de impedir el flujo de una corriente eléctrica. La resistividad de una formación viene dada por la naturaleza, cantidad y distribución del agua contenida en dicha formación. ^[4]

Registro de resonancia magnética. Se basa en la medición del tiempo de decaimiento o tiempo necesario para que núcleos de hidrogeno del agua e

hidrocarburos, cuya condición de equilibrio ha sido alterada a través de un campo magnético, pierdan energía y regresen a su estado inicial.^[4]

2.12.3 Registro para la medición del diámetro del hoyo

Registro Caliper. Mide de manera continua el diámetro del pozo en función de la profundidad. El uso principal es calcular el volumen de cemento necesario para llenar el pozo hasta cierta profundidad, determinar exactamente el diámetro del pozo para ser usado en la interpretación de otros registros y adicionalmente se pueden detectar zonas permeables por la presencia de zonas filtradas y localizaciones de centralizadores.^[4]

2.12.4 Registro para la medición de porosidad

Registro de velocidad acústica. Este se basa en la medida del tiempo de propagación de las ondas de sonido, este tiempo es proporcional a la porosidad de las rocas.^[4]

Registro de densidad. Este se basa en la medición del número de electrones que pueden transmitirse de una fuente de rayos gamma a un receptor, el número de electrones es proporcional a la densidad del medio. La relación entre la densidad y la porosidad es inversa.^[4]

Registro Neutrón. Este se basa en la medición de la radiación inducida en la formación por la pérdida de energía que sufren los neutrones con los que se bombardea la misma, esta pérdida de energía es proporcional a la porosidad.^[4]

2.13 Proceso de Ingeniería de Software

El proceso de Ingeniería del Software son etapas en las que las necesidades del usuario son traducidas en requerimientos del software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para uso operativo. Concretamente define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases.^[11]

Concepción: Define el alcance del proyecto y desarrolla un caso de negocio.

Elaboración: Define un plan del proyecto, especifica las características y fundamenta la arquitectura.

Construcción: Crea el producto.

Transición: Transfiere el producto a los usuarios.

2.14 Fases de la Ingeniería de Software

La ingeniería de software está dividida en las siguientes fases:^[11]

2.14.1 Análisis y Especificación de los Requerimientos

Esta fase consiste en analizar, entender y registrar el problema que el patrocinante está tratando de resolver. Los requerimientos son una descripción de las necesidades o deseos de un producto. Un conjunto de requerimientos en estado de madurez, debe presentar una serie de características tanto individualmente como en grupo, así se tiene que un requerimiento debe ser:^[11]

Necesario: Si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o proceso.

Conciso: Si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

Completo: Si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

Consistente: Si no es contradictorio con otro requerimiento.

No ambiguo: Cuando tiene una sola interpretación.

2.14.2 Diseño

Consiste en crear una solución que satisfaga las especificaciones definidas en la fase de análisis. En esta fase se crea la interfaz de usuario y la estructura del software.
[11]

2.14.3 Codificación

En esta fase se lleva a cabo la implementación física de las bases de datos, de los programas (codificación), prueba de unidad o procedimientos separados y prueba de subsistemas o integración.^[11]

2.14.4 Prueba del Sistema

Los grupos de procedimientos de unidades probadas como elementos separados en la fase previa son comprobados como sistema durante la integración.^[11]

2.14.5 Instalación

Corresponde a la fase de puesta en marcha de la aplicación. ^[11]

2.14.6 Mantenimiento del Sistema y Mejoras

Esta fase se corresponde al mantenimiento y mejoras futuras que deben contemplarse a la hora de implantar una aplicación informática, una vez que el sistema está en funcionamiento. ^[11]

2.15 Ciclo de vida del software

Al igual que en otros sistemas de ingeniería, los sistemas de software requieren un tiempo y esfuerzo considerable para su desarrollo y deben permanecer en uso por un periodo mucho mayor. Durante este tiempo de desarrollo y uso, desde que se detecta la necesidad de construir un sistema de software hasta que este es retirado, se identifican varias etapas que en conjunto se denominan el ciclo de vida del software. En función de cuales sean las características del proyecto, se configurará el ciclo de vida de forma diferente. Usualmente se consideran las etapas: especificación y análisis de requisitos, diseño del sistema, implementación del software, aplicación y pruebas, entrega y mantenimiento. Un aspecto esencial dentro de las tareas del desarrollo del software es la documentación de todos los elementos y especificaciones en cada fase. Dado que esta tarea siempre estará influida por la fase del desarrollo en curso, se explicará de forma distribuida a lo largo de las diferentes fases; como un apartado especial, para recalcar su importancia en el conjunto del desarrollo del software. ^[5]

Las etapas principales a realizar en cualquier ciclo de vida son:

Análisis: Construye un modelo de los requisitos

Diseño: A partir del modelo de análisis se deducen las estructuras de datos, la estructura en la que descompone el sistema y la interfaz de usuario.

Codificación: Construye el sistema. La salida de esta fase es código ejecutable.

Pruebas: Se comprueba que se cumplen criterios de corrección y calidad.

Mantenimiento: En esta fase, que tiene lugar después de la entrega se asegura que el sistema siga funcionando y adaptándose a nuevos requisitos.

Las etapas constan de tareas. La documentación es una tarea importante que se realiza en todas las etapas. Cada etapa tiene como entrada uno o varios documentos procedentes de las etapas anteriores y produce otros documentos de salida según se muestra en la figura 2.2. ^[5]

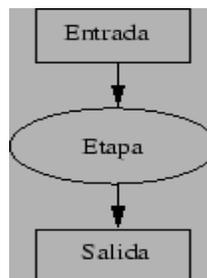


Figura 2.2 Etapa genérica. Fuente Ferre, G.

Algunos autores dividen la fase del diseño en dos partes: *Diseño global* o arquitectónico y *diseño detallado*. En el primero se transforman los requisitos en una arquitectura de alto nivel, se definen las pruebas que debe satisfacer el sistema en su conjunto, se esboza la documentación y se planifica la integración. En el detallado para cada módulo se refina el diseño, se definen los requisitos del módulo y su documentación.

Las formas de organizar y estructurar la secuencia de ejecución de las tareas en las diferentes fases de cada uno de los métodos puede dar lugar a un tipo de ciclo de

vida diferente. Los principales ciclos de vida que se van a presentar a continuación realizan estas tareas. Cada uno de ellos tiene sus ventajas e inconvenientes.

2.15.1 Ciclos de vida en cascada

El ciclo de vida inicialmente propuesto por Royce en 1970, fue adaptado para el software a partir de ciclos de vida de otras ramas de la ingeniería. Es el primero de los propuestos y el más ampliamente seguido por las organizaciones (se estima que el 90% de los sistemas han sido desarrollados así). La estructura se muestra en la Figura 2.3.^[5]

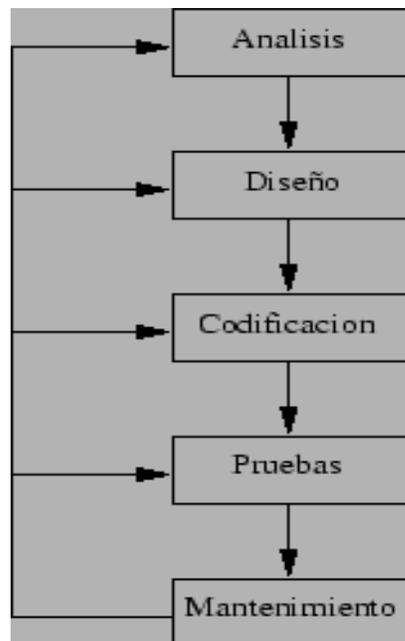


Figura 2.3 Ciclo de vida en cascada. Fuente Ferre, G.

2.15.2 Descripción del Ciclo de vida en Cascada

Este modelo admite la posibilidad de hacer iteraciones, es decir, durante las modificaciones que se hacen en el mantenimiento se puede ver por ejemplo la necesidad de cambiar algo en el diseño, lo cual significa que se harán los cambios

necesarios en la codificación y se tendrán que realizar de nuevo las pruebas, es decir, si se tiene que volver a una de las etapas anteriores al mantenimiento hay que recorrer de nuevo el resto de las etapas. ^[5]

Después de cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente.

Trabaja en base a documentos, es decir, la entrada y la salida de cada fase es un tipo de documento específico. Idealmente, cada fase podría hacerla un equipo diferente gracias a la documentación generada entre las fases. Los documentos son:

Análisis: Toma como entrada una descripción en lenguaje natural de lo que quiere el cliente. Produce el S.R.D. (Software Requirements Document).

Diseño: Su entrada es el S.R.D. Produce el S.D.D. (Software Design Document).

Codificación: A partir del S.D.D. produce módulos. En esta fase se hacen también pruebas de unidad.

Pruebas: A partir de los módulos probados se realiza la integración y pruebas de todo el sistema. El resultado de las pruebas es el producto final listo para entregar.

CAPÍTULO III

ANÁLISIS Y ESPECIFICACIÓN DE LOS REQUERIMIENTOS

3.1 Introducción

En este capítulo se desarrollara la primera etapa del ciclo de vida del software, conocida como la etapa de análisis de requerimientos, en donde se definirá el sistema y se realizara un diseño general. De esta forma, se lograra comprender el alcance, a través de sus funciones, los requisitos y actores que intervienen en el proceso y todo esto con la finalidad de desarrollar modelos preliminares que representen el comportamiento del sistema. En esta fase se definirán los flujos de trabajo, requisitos y funciones del sistema.

Esta fase se elaborara con la finalidad de obtener mediante de la recolección y análisis de información, los pasos fundamentales con los cuales se desarrolla el proceso, así como los requerimientos de los usuarios para el sistema, y de esta forma poder modelar y estructurar un software, efectivo, eficiente, flexible, amigable, y dinámico al usuario final.

La elaboración del proceso que cumplirá con el sistema, se logro después de analizar la información recopilada, y de esta forma generar un bosquejo del sistema muy general, ya que la función de la etapa de análisis no es modelar el sistema en si, sino, es bosquejar o tener una opinión racional del potencial sistema, y decidir si es viable invertir en un estudio más profundo.

3.2 Estudio del contexto del sistema

En todo proyecto de desarrollo de software se debe tener una perspectiva del ambiente donde se piensa implementar el sistema. En el presente proyecto se llevo a cabo un análisis, de las actividades realizadas por la GODD (Gerencia de Operaciones De Datos) de la empresa PDVSA distrito guaraguao, con el propósito de comprender el proceso que conlleva el estudio de registros obtenidos de un pozo de extracción de petróleo determinado, para su posterior análisis y almacenamiento, para cumplir dicho proceso, y poder esquematizar y diseñar el sistema. Para lograr este propósito fue necesario lo siguiente:

Realizar entrevistas al personal de la GODD (Gerencia de Operaciones De Datos) encargado del análisis y almacenamiento de los registros.

Llevar a cabo una depuración de los requerimientos del personal de la GODD, a fin de establecer los límites y funciones del sistema.

Analizar, los registros de pozos obtenidos, matemáticos y gráficos a realizar en cada uno de ellos.

3.2.1 Proceso de la GODD

La figura 3.1 muestra las actividades que se desarrollan así como los entes que intervienen en el desarrollo de un análisis y almacenamiento de datos.

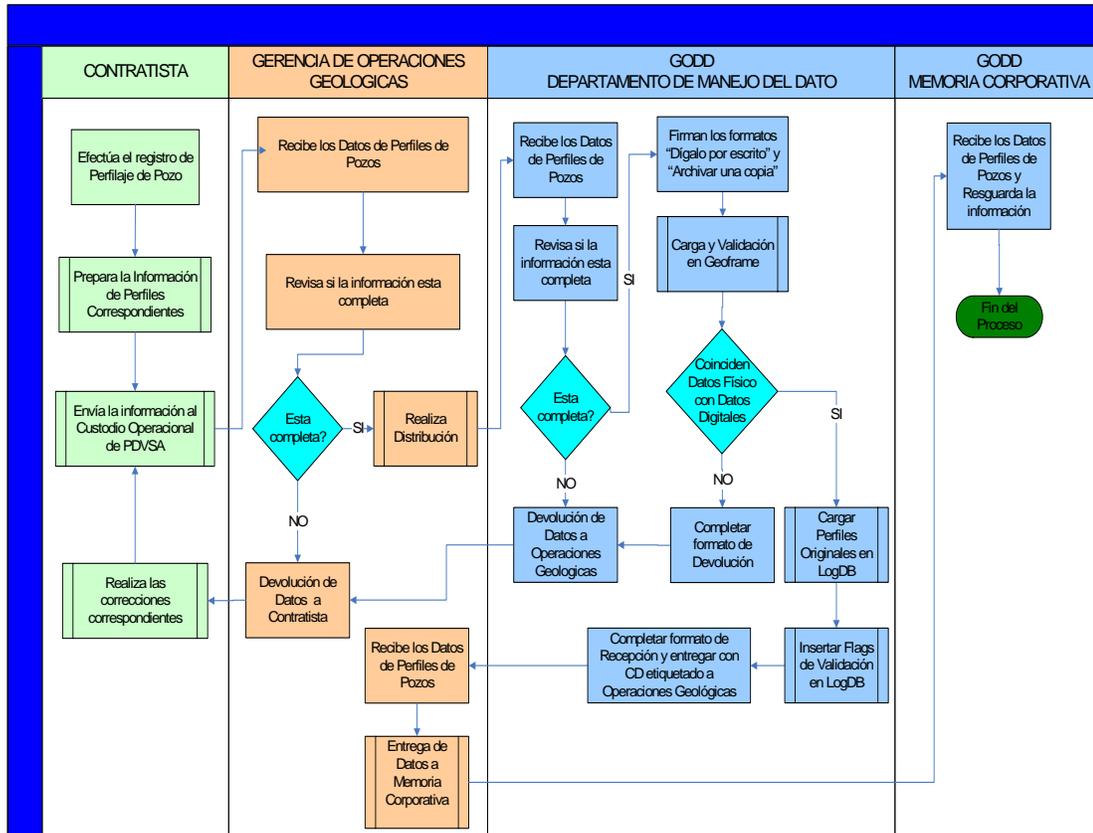


Figura 3.1 Flujo de Trabajo de la GODD. Fuente Propia

Este proceso comienza en la empresa contratada por Petróleos de Venezuela S.A., con la realización del registro de perfilaje de pozo donde se encuentran todos los datos, luego se separa la información, agrupando los datos para ser enviados al custodio operacional de la empresa PDVSA.

La gerencia de Operaciones Geológicas, recibe el paquete de datos y revisa la información. En caso de que los datos no estén completos, se realiza la devolución a la empresa contratada para que esta haga las correcciones pertinentes, para ser enviados nuevamente a la empresa. En caso contrario, la Gerencia de Operaciones Geológicas realiza la distribución de los datos.

En la Gerencia de Operaciones de Datos (GODD), departamento de Manejo de Datos, la información es recibida y de nuevo revisada, si los datos no están completos se devuelven a la contratista, de otro modo se firman los formatos “Dígalo por Escrito” y “Archiva una copia”, para proceder a cargar y validar los datos en Geoframe. En el proceso de validación se revisa que los datos físicos entregados correspondan con los datos digitales, si estos no corresponden, se devuelven a la gerencia de Operaciones Geológicas, para que ellos tomen las medidas pertinentes con la información.

Con los datos validados y cargados en Geoframe, se procede a cargar los perfiles originales en el software LogDB, se insertan los flags (etiquetas) de validación en los perfiles y se mandan a la Gerencia de Operaciones Geológicas, donde se quedan con una copia y mandan la que se va a archivar en GODD departamento de Memoria Corporativa.

3.2.2 Roles y responsabilidades

GODD manejo de datos:

Es el ente encargado de la recepción y análisis de los datos de los perfiles de pozos (registros).

Entre sus responsabilidades se encuentran:

- Revisar si la información de los registros esta completa.
- Firmar los formatos “Dígalo por escrito” y “Archivar una copia”.
- Cargar y validar información en Geoframe (software de graficación bajo sistema operativo UNIX).

- Cargar perfiles originales e insertar los flags (etiquetas) de validación en LogDB (software de almacenamiento de los datos de los perfiles bajo sistema operativo Windows).
- Completar el formato de recepción y entregar un disco al ente operaciones geológicas.
- Si la información esta incompleta devolverla al ente operaciones geológicas.
- Completar el formato de devolución

GODD memoria corporativa:

Ente encargado del almacenamiento de la información, para su posterior utilización consulta o estudio en el comportamiento de un pozo determinado.

Su responsabilidad es:

- Resguardar la información de los perfiles de pozos.

3.2.3 Modelo de Dominio

Utilizando la notación UML, un modelo del dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Pueden mostrar objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y atributos de las clases conceptuales.

Las clases del dominio más importantes que se observan (ver figura 3.2) son: Operaciones_Geológicas, GODD_Memoria_Corporativa, GODD_Manejo_de_Dato. En este modelo el analista puede comparar, verificar, y clasificar la data de los

perfiles de pozos, así como devolverla en dato caso de que sea errónea o carezca de información, así como realizar la graficación computarizada y estudio de los datos de los perfiles de pozos, y posteriormente de la verificación de datos devolver un respaldo al ente Operaciones_Geológicas.

En el modelo de dominio de la figura 3.2 se puede observar la interacción de los diferentes objetos o entes llevan acabo el flujo de actividades de toma, resguardo, y almacenamiento de la data, en donde el objeto Operaciones Geológicas envía el archivo con los datos de los perfiles de pozos, el cual es recibido por el objeto GODD_Manejo_de_Datos el cual verifica los datos del archivo y compara si el registro digital concuerda con el registro físico, si concuerdan el objeto GODD_Manejo_de_Datos, realizara una copia de los datos y los enviara al ente o objeto GODD_Memoria_Corporativa, el cual se encargara de su resguardo y envió de formato de aceptación de datos y recibo de los mismos. En caso de que la data sea errónea, el objeto GODD_Manejo_de_Datos, los enviara al objeto Contratista para su corrección.

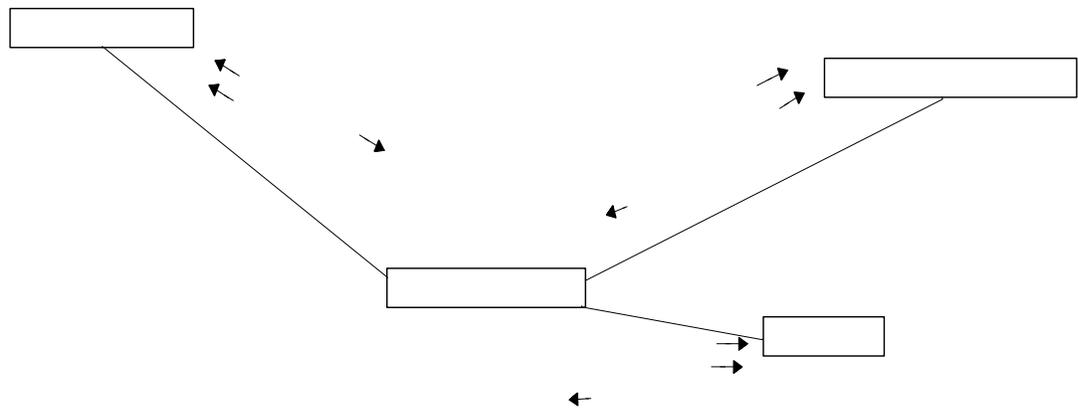


Figura 3.2 Modelo de Dominio. Fuente Propia

3.2.4 Glosario de términos del Modelo de Dominio

Tabla 3.1 Glosario de términos del modelo de Dominio. Fuente propia

| TERMINO | DEFINICIÓN |
|--------------------------|--|
| Operaciones_Geologicas | Ente cuyo personal se encarga del envío de la data y recepción de la data y graficas de perfiles de pozos. |
| GODD_Manejo_de_Datos | Departamento encargado de la verificación, comparación, validación, y graficación de los datos de los perfiles de pozos. |
| GODD_Memoria_Corporativa | Departamento donde su personal se encarga del almacenamiento, de los datos de los perfiles de pozos producibles |

3.3 Riesgos del Sistema

El principio general es tener un plan de acción a seguir sobre los riesgos. Las fases y las iteraciones dentro de las fases proporcionan el medio de planificar las acciones sobre los riesgos.

3.3.1 Riesgos Críticos del Sistema

- Desconocimiento del ámbito del sistema

Descripción: El no conocer el ámbito del sistema en base al cual se está trabajando, significa un riesgo crítico que debe ser mitigado en la fase de inicio. El

diseñador debe tener un conocimiento del contexto y entender sus aspectos fundamentales.

Responsable: Analistas y Diseñadores del Sistema.

- Requisitos mal definidos

Descripción: No disponer de unos requisitos claros y que abarquen todas las necesidades de los usuarios, pueden significar un riesgo para el sistema. Se deben recolectar todas las necesidades de los usuarios.

Responsable: Analistas y Diseñadores de Sistemas.

- Falta de robustez de la arquitectura

Descripción: Un riesgo crítico que debe ser mitigado en la fase de inicio es el riesgo de falta de robustez de la arquitectura. El sistema debe poseer una arquitectura robusta que le permita adaptarse a los cambios y al mantenimiento de manera elegante y poco traumática.

Responsable: Analistas y Diseñadores del Sistema.

3.4 Requisitos

3.4.1 Requisitos Funcionales

El sistema debe contar con una interfaz de fácil manipulación por parte del usuario.

La interfaz le debe permitir al usuario, generar tantas curvas sean necesarias, para un estudio, así como el tipo de curva a realizar.

Las graficas deben permitir representar los tipos de los cambios así como ajustar la escala de medición sin necesidad de hacer otro procedimiento.

El sistema debe permitir a los usuarios la selección y graficación de un número variado de curvas.

El sistema debe proveer al usuario de la información y la data del estudio a realizar.

3.4.2 Requisitos no funcionales

El sistema debe estar realizado con herramientas de Software Libre cumpliendo con el decreto 3.390, a manera de reducir los costos de desarrollo y su código pueda ser modificado sin ninguna restricción.

El sistema debe estar diseñado con una arquitectura que pueda adaptarse fácilmente a cualquier cambio y mejora estructural.

3.4.3 Requisitos de Software

- Sistema operativo: Linux Kernel 2.4 o superior, Windows 2000/XP/Vista.
- Adobe Acrobat 7.0 o superior.
- Microsoft Office XP/2003/2007 o OpenOffice.

3.4.4 Requisitos de la Plataforma Hardware

- 1 procesador x86 o equivalente a 1GHz o más.
- 512 MB de memoria de acceso aleatorio (RAM).
- Disco Duro de 40 Gb.
- Monitor a color con una resolución máxima de 1280 x 1024.
- Tarjeta grafica 128 MB

3.4.5 Identificación de los Actores

Tabla 3.2 Tabla de identificación de actores. Fuente Propia

| Actores | Descripción |
|----------------|---|
| Usuario | Es el encargado de la manipulación y control del software, así como la selección de la data, carga y graficación de los datos de los perfiles de pozo, dentro del departamento de operaciones de datos. |
| Archivo LAS | Encargado de proveer la información correspondientes a los datos de los pozos en digital al sistema. |
| Archivo Imagen | Encargado de proveer la información correspondientes a los datos de los pozos en Formaton de imagen digital al sistema. |

3.5 Caso de uso del sistema

En la figura 3.3 se puede observar un boosquejo general del caso de uso principal del sistema JGrafep.

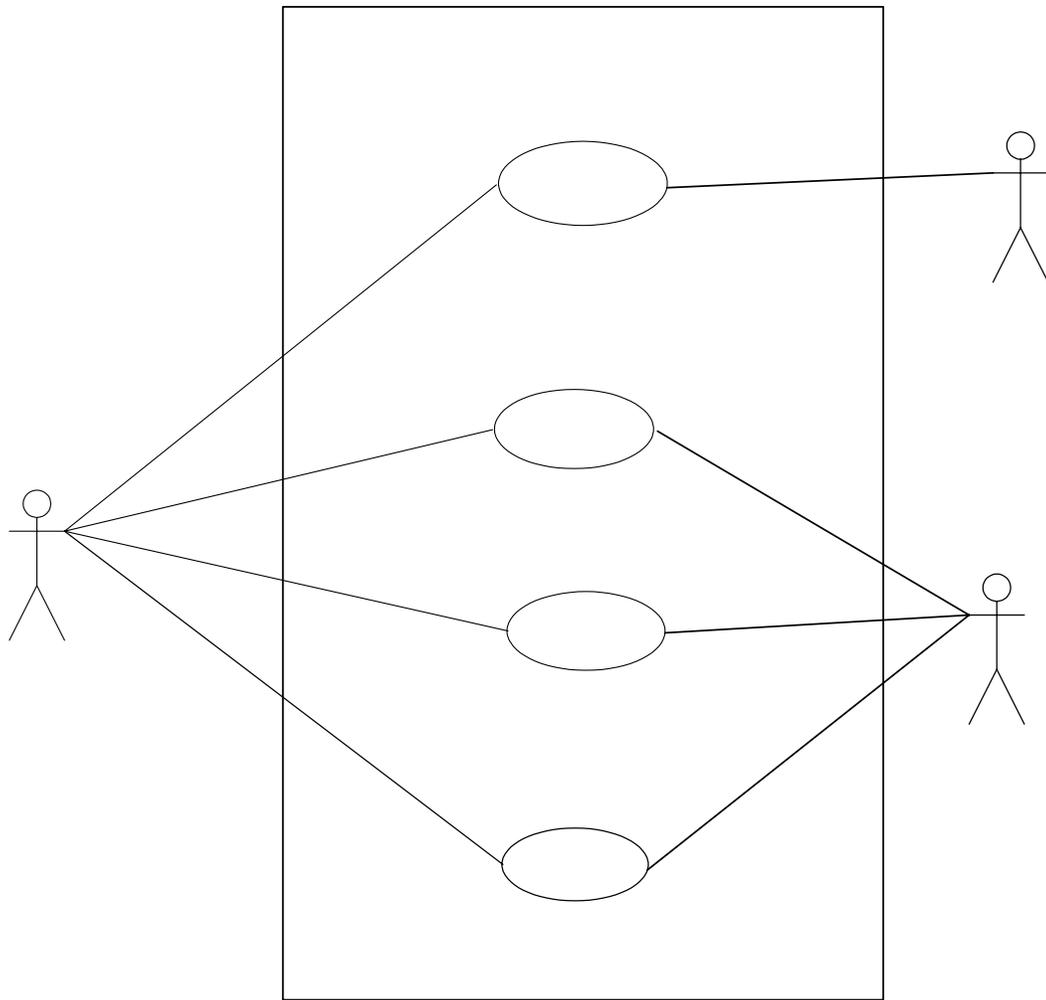


Figura 3.3 Caso de uso Principal del Sistema JGrafep. Fuente Propia

3.5.1 Caso de uso Mostrar Registro Datos

En este caso de uso el usuario accede al sistema para visualizar los registros de datos; para lo cual, el programa hace uso del manejador de archivos, en donde localiza al archivo, lo valida y lo lee; para luego mostrarlo al usuario.

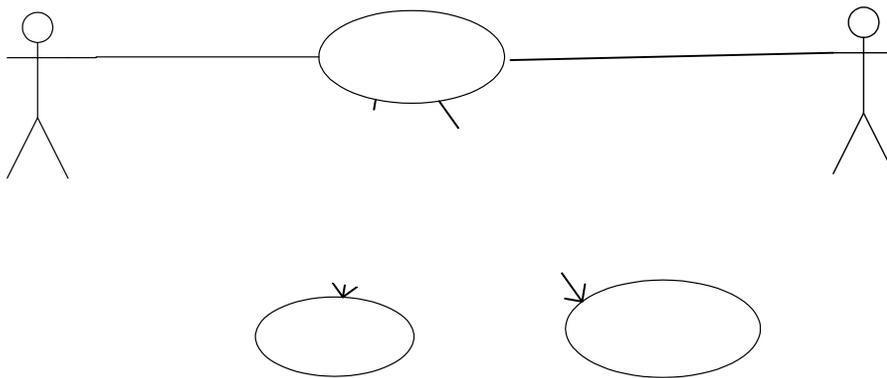


Figura 3.4 Caso de uso Mostrar Registro Datos del Sistema JGrafep. Fuente Propia

Para este caso de uso se consideraran la siguiente descripción:

Nombre del Caso de Uso: Mostrar Registro Datos.

Actores: Usuario y Archivo LAS

Descripción: En este caso de uso el usuario solicita al programa la apertura y mostrado por pantalla de un archivo de datos correspondiente al perfil de un pozo.

Pre-Condición: El usuario invoca el caso de uso.

Mostrar Registro
Datos

Usuario

Manejador de
Archivos

<<Includes>>

>>Includes

Flujo Principal:

Se despliega la interfaz que permite al usuario seleccionar, el archivo LAS que desea visualizar.

Finaliza el caso de uso.

Flujo alterno:

El usuario puede salir del programa en cualquier momento.

El usuario puede cerrar las interfaces con los datos y abrir otras interfaces con otros datos.

3.5.2 Caso de uso Mostrar Registro Imagen

El usuario accede al Sistema para visualizar los registros de imagen; para lo cual, el programa hace uso del manejador de archivos, en donde localiza al archivo, lo valida e inmediatamente carga la imagen; para luego mostrarlo al usuario.

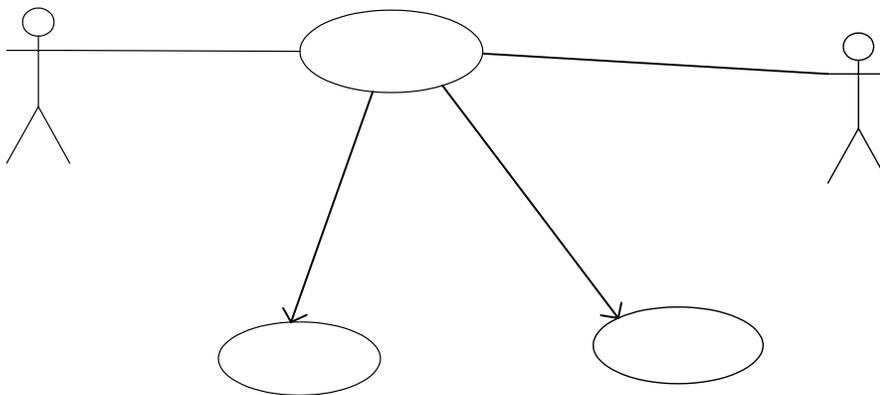


Figura 3.5 Caso de uso Mostrar Registro Imagen del Sistema JGrafep. Fuente Propia

Para este caso de uso se consideraran la siguiente descripción:

Nombre del Caso de Uso: Mostrar Registro Imagen.

Actores: Usuario y Archivo Imagen

Descripción: En este caso de uso el usuario solicita al programa la apertura y mostrado por pantalla de una imagen correspondiente al perfil de un pozo.

Pre-Condición: El usuario invoca el caso de uso.

Flujo Principal:

Se despliega la interfaz que permite al usuario seleccionar, el archivo Imagen que desea visualizar.

Finaliza el caso de uso.

Flujo alterno:

El usuario puede salir del programa en cualquier momento.

El usuario puede cerrar la interfaz con la imagen y abrir otra interfaz con otra imagen.

3.5.3 Caso de uso Graficar Curva

El usuario accede al sistema para graficar curvas. Para lo cual, el programa realiza la selección del archivo haciendo uso del manejador de archivos, en donde

localiza al archivo, lo valida e inmediatamente carga los datos. El programa selecciona las curvas, selecciona los valores de los rangos de las curvas y carga los datos de las mismas. Finalmente muestra la grafica al usuario; creando las leyendas, rótulos y dibujando las curvas.

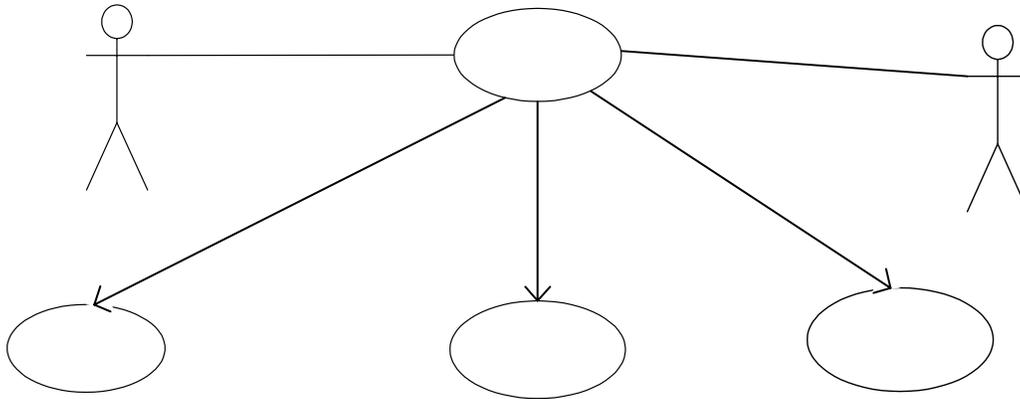


Figura 3.6 Caso de uso Graficar Curva del Sistema JGrafep. Fuente Propia

Para este caso de uso se consideraran la siguiente descripción:

Nombre del Caso de Uso: Graficar Curva.

Actores: Usuario y Archivo LAS

Descripción: En este caso de uso el usuario solicita al programa la creación de una o varias graficas de perfil de pozos, obteniendo los datos del Archivo LAS.

Usuario

Pre-Condición: El usuario invoca el caso de uso.

Flujo Principal:

Seleccionar

Archivo

Se despliega la interfaz que permite al usuario seleccionar, el archivo que desea graficar.

El usuario selecciona la escala de la grafica, su tipo y las curvas que desea graficar.

El usuario establece la posición de la grafica, su color y sus rangos (derecho e izquierdo).

Se presiona el botón graficar y el programa despliega una interfaz con el resultado de la grafica.

Finaliza el caso de uso.

Flujo alterno:

El usuario puede salir del programa en cualquier momento.

El usuario puede cerrar la interfaz con la gráfica y comenzar otra gráfica.

3.5.4 Caso de uso Editar Archivo Datos

El usuario accede al sistema para editar los archivos de datos. Se carga el archivo localizándolo, validándolo y generando una vista en pantalla de los datos. Se guardan los cambios realizados; se muestra el resultado y se efectúa el almacenamiento de los datos a través del manejador de archivos.

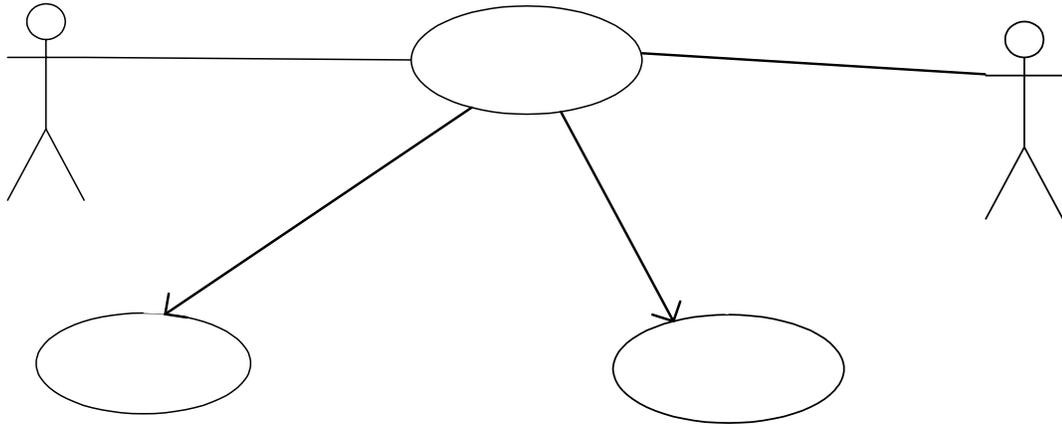


Figura 3.7 Caso de uso Editar Archivo Datos del Sistema JGrafep. Fuente Propia

Para este caso de uso se consideraran la siguiente descripción:

«include»

Nombre del Caso de Uso: Editar Archivo Datos.

Usuario

Actores: Usuario y Archivo LAS

Descripción: En este caso de uso el usuario solicita al programa la creación de un archivo de datos, en base a otro u otros archivos de datos.

Cargar Archivo

Pre-Condición: El usuario invoca el caso de uso.

Flujo Principal:

Se despliega la interfaz que permite al usuario seleccionar, el archivo que se mostrara como archivo Origen.

En la misma la interfaz el usuario selecciona el archivo que se mostrara como archivo Destino.

El usuario realiza las labores de copiado y pegado en los archivos respectivos.

El usuario realiza la operación de almacenamiento de los datos, ya sea con un nombre nuevo o con el original del archivo.

Finaliza el caso de uso.

Flujo alterno:

El usuario puede salir del programa en cualquier momento.

El usuario puede cerrar la interfaz con los archivos de datos y comenzar otra edición de datos.

3.6 Análisis

Durante este flujo de trabajo, analizaremos los requisitos que se describieron en el flujo de trabajo anterior, refinándolos y estructurándolos.

El resultado del flujo de trabajo del análisis es el modelo de análisis, que es un modelo del objeto conceptual que analiza los requisitos mediante su refinamiento y estructuración.

3.6.1 Diagramas de clases de Análisis

Los diagramas de clases de análisis plasman las posibles clases del diseño, encajándolas en los tres estereotipos básicos: de interfaz, de control y de entidad. Cada uno de ellos implica una semántica específica, lo cual constituye un método consistente de identificar y describir cada clase, contribuyendo a la creación de un modelo de objetos y una arquitectura robusta.

3.6.2 Identificación de Clases de Análisis

Clases de Interfaz

Las clases de interfaz se utilizan para modelar la interacción entre el sistema y sus actores, esta interacción a menudo implica recibir información y peticiones de (y hacia) los usuarios y los sistemas externos.

Clases de Control

Las clases de control representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto.

Clases de Entidad

Las clases de entidad se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Las clases de entidad modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto o un suceso.

Tabla 3.3 Tabla de identificación de clases de control. Fuente Propia

| Clases de control del Sistema JGrafep | |
|---------------------------------------|---|
| Clase de Control | Descripción |
| Controlador Cargar Archivo | Encargado de la carga de la carga y validación de la los datos de los archivos de perfiles de pozos |
| Controlador Cargar Imagen | Encargado de realizar el ajuste y verificación del tipo de imagen, valida para el sistema. |
| Controlador Graficar | Se encarga del ordenamiento de los datos a graficar según la selección de registros realizada del operador. |
| Controlador Editar Archivo | Encargado de realizar la carga del archivo así como la edición del mismo. |

Tabla 3.4 Tabla de identificación de clases de interfaz. Fuente Propia

| Clases de control del Sistema JGrafep | |
|---------------------------------------|---|
| Clase de Interfaz | Descripción |
| Interfaz Cargar Archivo | Estereotipo que representa la interfaz de carga de archivo, que le permite usuario seleccionar un archivo LAS específico. |
| Interfaz Principal | Estereotipo que representa la interfaz principal o de entrada al sistema, donde se realiza la selección de una función específica: Modificar, graficar, cargar imagen; entre otras. |
| Interfaz Cargar Imagen | Estereotipo que representa la interfaz que permite realizar la búsqueda de la imagen a mostrar por pantalla- |
| Interfaz Datos Grafica | Estereotipo que representa la interfaz que permite la carga y selección de los registros de pozos y sus características para ser graficados |
| Interfaz Grafica | Estereotipo que representa la interfaz que permite la visualización de las diferentes curvas seleccionadas por el operador así como sus diferentes escalas y colores. |
| Interfaz Editar Archivo | Estereotipo que representa la interfaz mediante la cual seleccionar el archivo a ser modificado. |

Tabla 3.5 Tabla de identificación de clases de Entidad. Fuente Propia

| Clases de control del Sistema JGrafep | |
|---------------------------------------|--|
| Clase Entidad | Descripción |
| Archivo | Estereotipo que representa el objeto encargado del almacenamiento temporal del archivo de los registro de pozos. |
| Imagen | Estereotipo que representa el objeto encargado del almacenamiento de los diferentes tipos de imágenes para su posterior visualización. |
| Archivo de Datos | Estereotipo que representa el objeto encargado del ordenamiento de los datos de las curvas según su tipo de grafica y escala. |

3.6.3 Diagrama de clases de análisis Cargar Archivo de Datos

El usuario activa la “interfaz_Cargar Archivo” del sistema y selecciona la data a cargar.

Los datos son cargados y verificados por el controlador “Cargar archivo”, el cual depura los archivos.

Los datos son cargados en el manejador de archivos.

Los datos depurados y clasificados son enviados al “Controlador Cargar archivo”.

Los datos son cargados en la “interfaz_Cargar archivo” para ser mostrados al usuario.

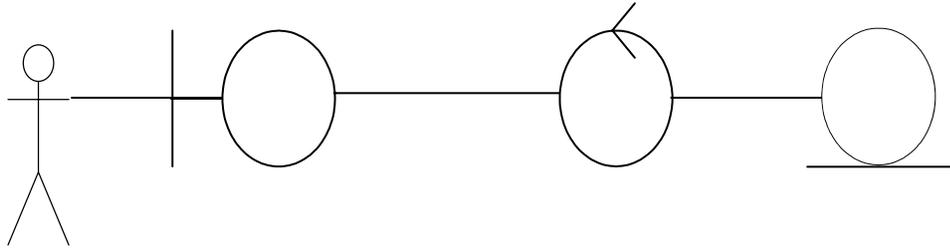


Figura 3.8 Diagrama de clases de análisis Cargar Archivo de Datos. Fuente propia

3.6.4 Diagrama de clases de análisis archivo de Imagen

El usuario realiza la activación de la “interfaz_Cargar Imagen” y selecciona la imagen a cargar.

La imagen es enviada al “Controlador Cargar Imagen” para su verificación y ajuste.

La imagen es enviada a la entidad “imagen” para ser cargada en el objeto específico.

La imagen es enviada al “Controlador Cargar Imagen” para realizar el procedimiento de muestra de la imagen.

La imagen es enviada a la “interfaz_Cargar Imagen” para su visualización por parte del usuario.

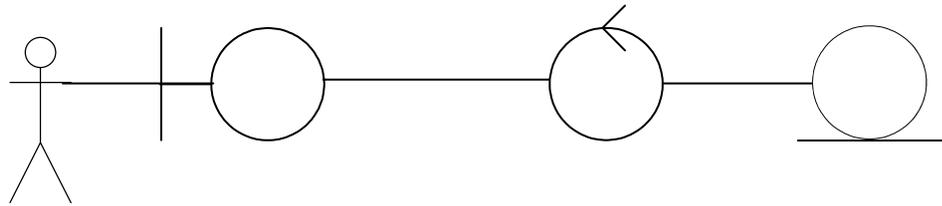


Figura 3.9 Diagrama de clases de análisis archivo de Imagen. Fuente propia

3.6.5 Diagrama de clases de análisis Graficar Curva

El usuario activa la interfaz “Interfaz_Principal”.

Los datos son cargados y verificados por el “Controlador Cargar archivo”, el cual depura los archivos.

Activa la interfaz “Interfaz_Datos Grafica” y selecciona los estudios a graficar, el tipo de curva, y la escala a graficar.

Los datos de los estudios a graficar son enviados al controlador “Controlador Graficar” en donde se realizara una verificación y depuración de la data enviada.

La entidad “Archivo de Datos” realiza un ordenamiento de los datos según el tipo de grafica seleccionada.

Los datos son enviados al controlador “Controlador Graficar” para su ordenamiento en la interfaz.

Los datos son enviados a la interfaz “Interfaz_Grafica” en donde se procederá a la ubicación de las curvas elaboradas.

La interfaz “Interfaz_Datos Grafica” realiza una activación del controlador “Controlador Graficar” y esterealiza la activación y envía los datos a la interfaz “Interfaz Grafica” para realizar la visualización de la curvas por parte del usuario.

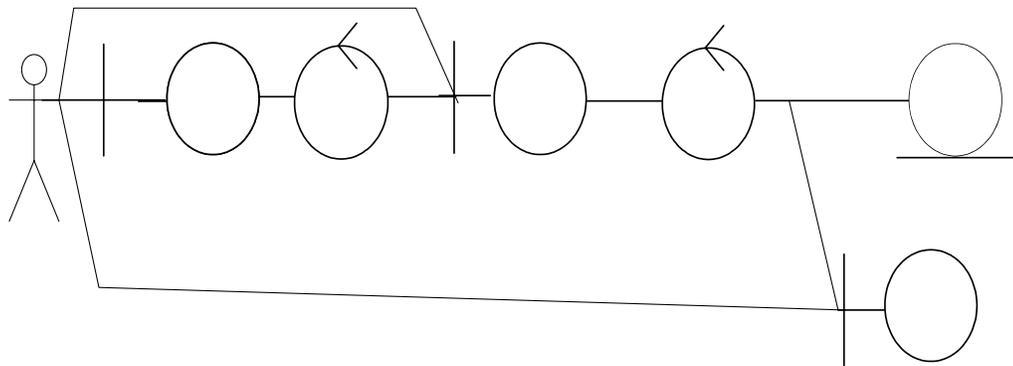


Figura 3.10 Diagrama de clases de análisis Graficar Curva. Fuente propia

3.6.6 Diagrama de clases de análisis Editar Archivo Datos

El usuario activa la interfaz y selecciona la “Interfaz_Editar” Archivo.

Realiza una búsqueda del archivo a través del controlador “Editar Archivo”.

El archivo de datos de los estudios de perfiles de pozos es cargado por la entidad “Archivo de Datos” para su posterior edición.

Usuario

El archivo de datos es enviado al controlador “Editar Archivo” para realizar las operaciones de edición.

Los datos del archivo son enviados a la interfaz “Editar Archivo” para la visualización del usuario.

Interfaz Principal Controlador
Cargar archivo

Interfa

Los datos son seleccionados y enviados al controlador “Editar Archivo”. y posteriormente, el archivo de estudio de perfiles de pozos es editado, validado por el controlador “Editar archivo”.

Los nuevos datos son guardados y por la entidad “Archivo de Datos”.

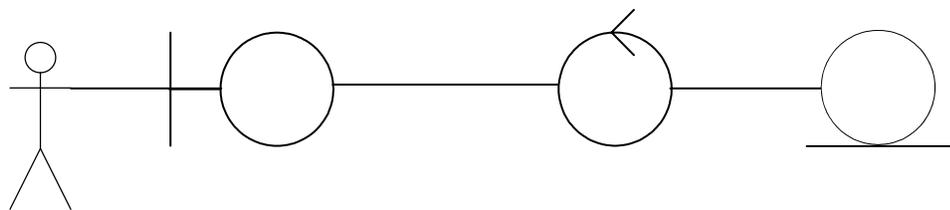


Figura 3.11 Diagrama de clases de análisis Editar Archivo Datos. Fuente propia.

Interfaz_Editado
Archivo

Usuario

CAPÍTULO IV

DISEÑO

4.1 Fase de diseño

El objetivo principal de esta fase es alcanzar la línea base de la arquitectura, recopilando la mayoría de los requisitos que aún quedan pendientes. En esta fase se van a generar los modelos del programa con el propósito de desarrollar la solución más cercana al mundo real.

4.1.1 Diagrama de clases de colaboración para el caso de uso Graficar Curva

El diagrama muestra la interacción entre las diferentes clases que conforman el proceso Graficar. Comenzando con la Interfaz Principal desde la cual se efectúa la llamada a la clase Graficar, en donde se muestra al usuario las curvas disponibles para graficar y se ordena al programa efectuar la grafica indicada, esta clase también contiene a las clases abstractas InformacionCurvas e InformacionEncabezado, estas son utilizadas para generar la grafica deseada. La clase InformacionCurvas contiene la información de todas las curvas ha graficar, los datos de los estudio y sus nombres; esta clase es llamada por el método Graficar() de la clase Graficar. La clase InformacionEncabezado contiene la información del encabezado del archivo de curvas, las unidades de cada curva, el valor nulo y el valor de división. La clase InformacionCurvas crea una instancia de la clase Estudio en donde ordena los datos necesarios para elaborar la grafica seleccionada, conteniendo ésta los datos de la curva, el color de la grafica, el rango, el lado de la grafica de perfil de pozo, la escala y la unidad.

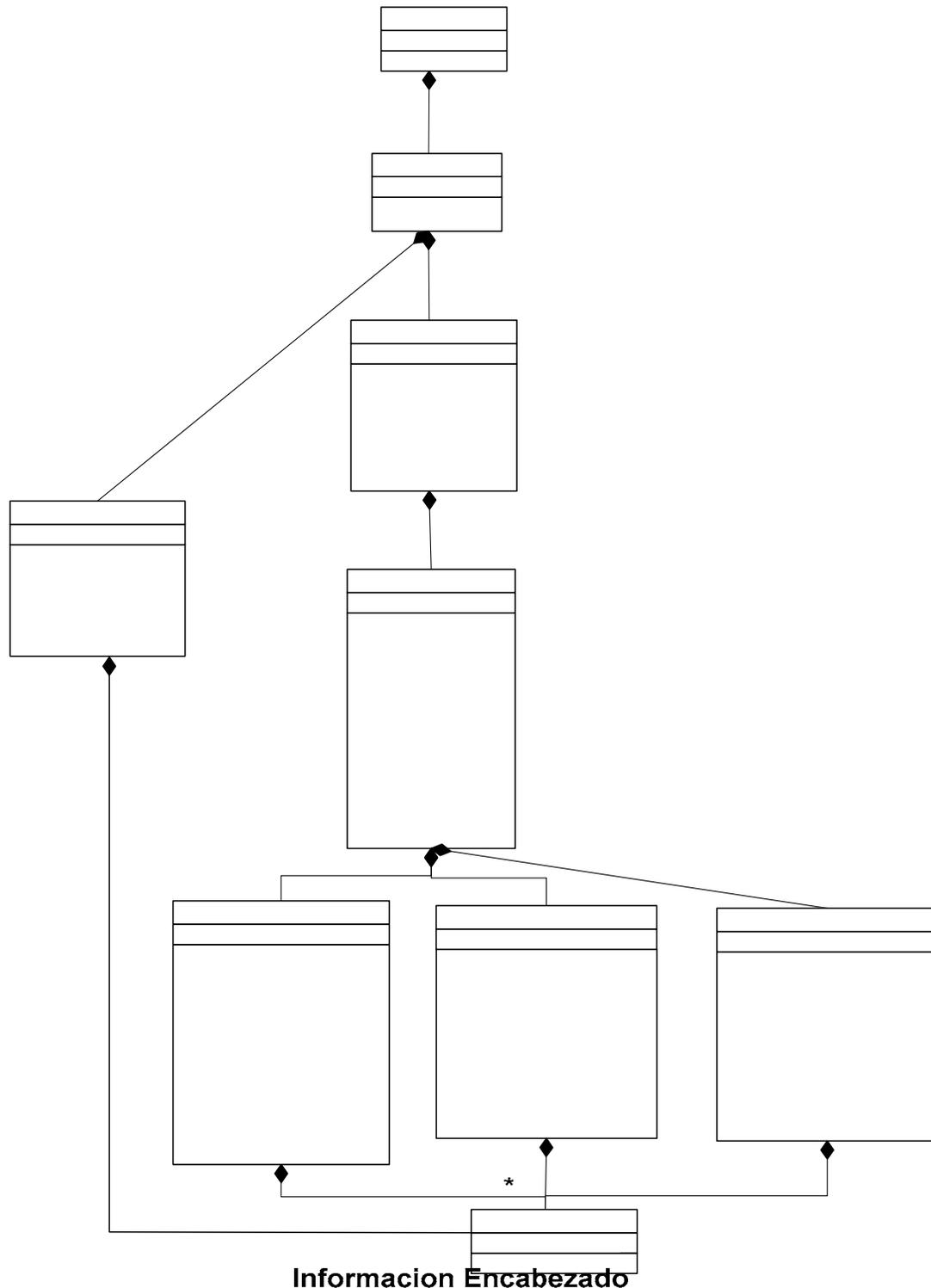


Figura 4.1 Diagrama de clases para el caso de uso Graficar Curva. Fuente Propia

- + Recibir_Archivo()
- + MostrarEncabezado()
- + ObtenerDivisio()
- + ObtenerEncabezado()
- + ObetenerNulo()
- + ObtenerUnidades()
- + TamañoEncabezado()

Interfaz

1

1

Graf

+ MostarC

+ Graficar

1

1

*

Informacio

+ AsignarNulo

+ BorrarCurva

+ LeerDatos()

+ ObtenerCo

+ ObtenerTitu

+ ObtenerEs

+ Obtener_Ti

+ Recibir_Arch

1

1

Interfaz Data

+ AsignarColo

+ AsignarNor

4.2 Diagramas de clases.

La elaboración de los diagramas de clases permite dar inicio a la estructura final del programa, partiendo del concepto modular hacia la del código del mismo. En esta sección se desarrollará el prototipo de diagrama de clases con sus respectivos atributos y procedimientos.

4.3 Diagrama de secuencia.

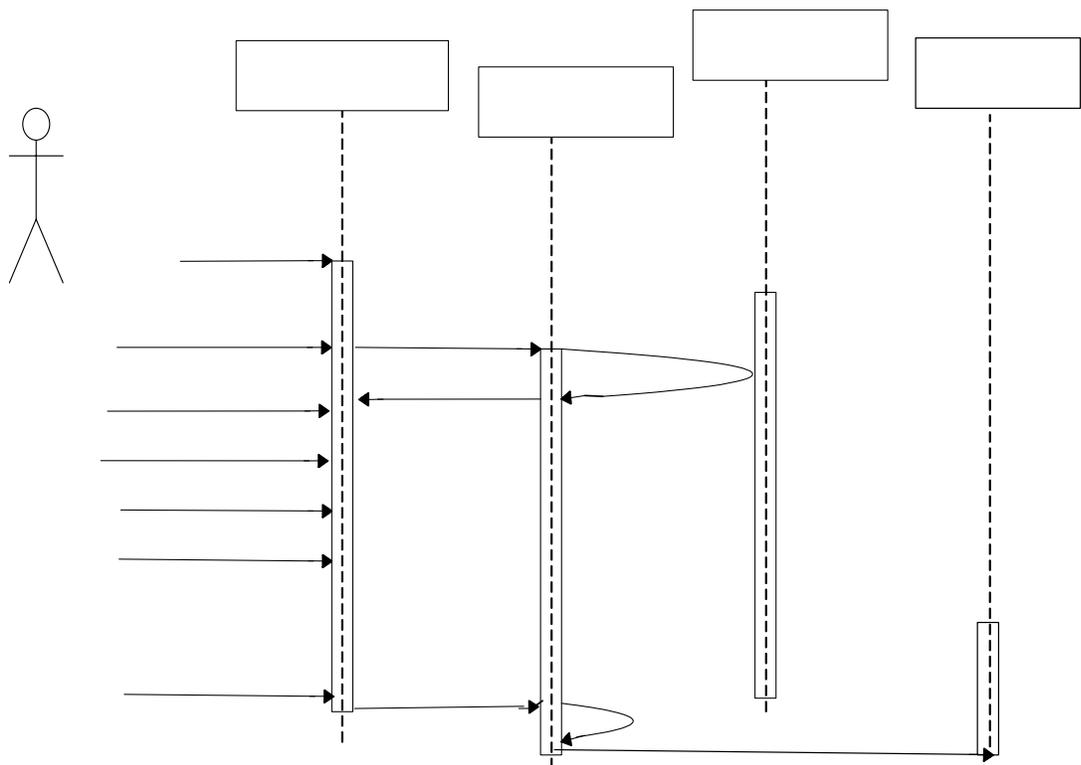


Figura 4.2 Diagrama de secuencia del proceso Graficar. Fuente propia.

En la figura 4.2 se puede observar que el usuario selecciona graficar de **Interfaz_Datos Grafica** un pozo determinado, esto crea una instancia de la interfaz grafica (Interfaz_Datos Grafica) que le permite al usuario desarrollar la grafica en el programa;

posteriormente se ingresa los datos del archivo a través de un paso de mensaje llamado Selección_de_Archivo este mensaje es procesado por el objeto Interfaz_Datos Grafica y le envía un mensaje (Archivo_Curvas) al objeto Archivo que la activa para procesar datos esta clase valida los datos y retorna un paquete (Archivo_Curvas) al objeto Interfaz_Datos Grafica; el objeto Interfaz_Datos Grafica recibe un mensaje Selección_de_Curvas, posteriormente un mensaje Seleccionar_Lado_Color, al tiempo recibe otro mensaje Seleccionar_Escala; en seguida es recibido otro mensaje Inserción_Rangos, por ultimo es recibido el mensaje Graficar_Curvas en el objeto Interfaz_Datos Grafica y este origina un paquete llamado Datos_Grafica que es enviado al objeto Archivo, este realiza una conversión de datos denotado en la figura por el mensaje Conversión_Datos y además se genera un paquete Datos_Curvas que activa al objeto Interfaz_Graficar.

4.4 Diseño de interfaces.

4.4.1 Interfaz Principal.

El modelo de interfaz principal del sistema contendrá un menú de opciones de Archivo, en el cual se podrán encontrar opciones como Cargar Archivo de Datos, Cargar Archivo de Imagen, Graficar y salir. Otra opción que se mostrara en el menú de opciones es la pestaña de Edición, que poseera Editar Archivo de Datos. En el menú de opciones también estará la pestaña ayuda que ofrecera al usuario la oportunidad de tener acceso al manual de usuario de la aplicación, así como tener acceso a la versión del programa entre otros. En el panel principal se contarán con tres etiquetas que tendrán adjuntas imágenes correspondientes a la Gerencia de Operaciones de Datos

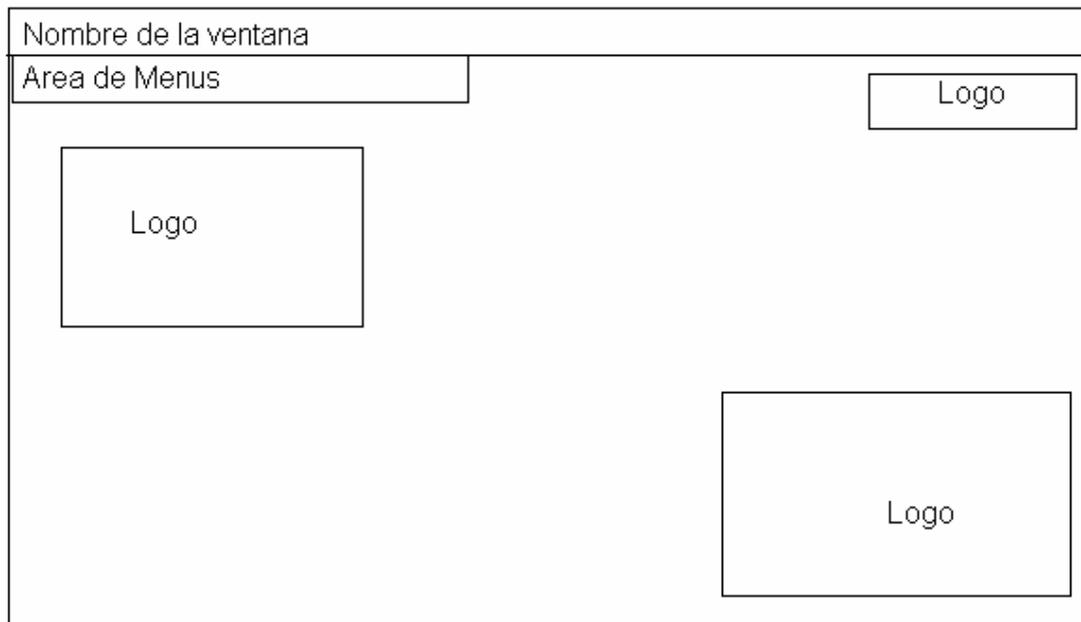


Figura 4.3 Bosquejo general de la interfaz principal. Fuente propia.

4.4.2 Interfaz Abrir Archivo de Datos.

La opción “Cargar Archivo de Datos” estará ubicada en la pestaña “Archivo” de la ventana principal del software; al hacer clic en dicha opción se desplegará una ventana correspondiente a un FileChooser, en el cual se hará un Filtrado de Archivos para que muestre por pantalla solo los archivos correspondientes a archivos de datos de perfiles de pozos (Archivos LAS).

Luego de la apertura del archivo, se mostrarán por pantalla dos nuevas ventanas, una correspondiente al encabezado del archivo, donde se muestran 5 casillas de activación, que pertenecen a los 5 bloques existentes dentro del encabezado del archivo. Mas abajo se muestra un campo de texto, con todos los bloques de datos seleccionados en las casillas de activación antes mencionadas, por defecto se configurará el programa para que muestre de primera instancia las opciones Versión e Información de pozo.

La otra ventana mostrara los datos correspondiente a todos los estudios realizados, estos seran mostrados por pantalla en un tabla que tendrá tantas columnas como estudios tenga el archivo.

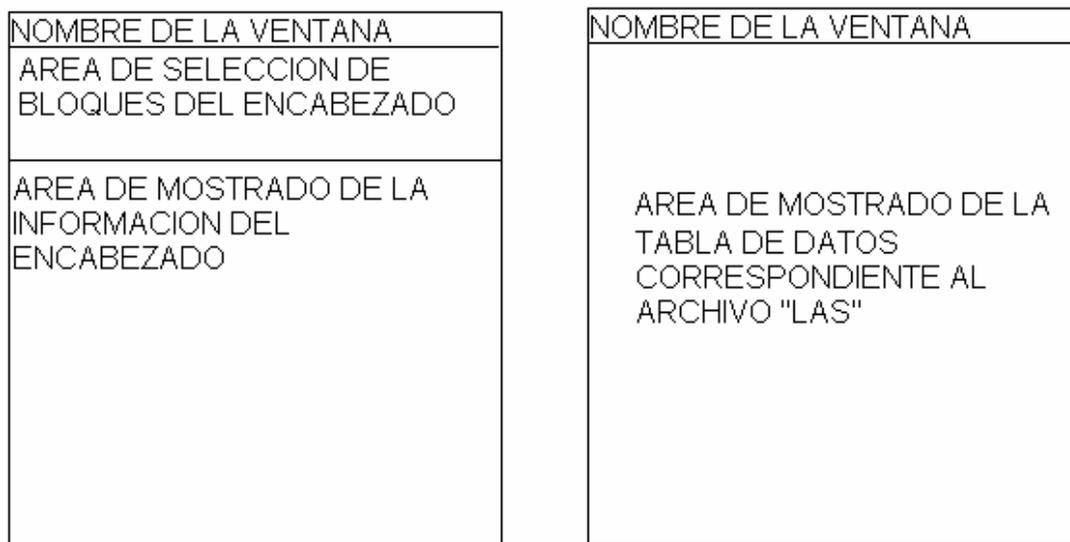


Figura 4.4 Bosquejo general de las interfaces de abrir archivo de datos. Fuente propia.

4.4.3 Interfaz Graficar.

La opción Graficar se encontrara en la pestaña “Archivo” de la ventana principal del software, al hacer clic en dicha opción se desplegara una ventana correspondiente a un FileChooser, en el cual se hace un Filtrado de Archivos para que muestre por pantalla solo los archivos correspondientes a archivos de datos de perfiles de pozos (Archivos LAS).

Luego de la apertura del archivo, se mostrara por pantalla una nueva ventana correspondiente a las opciones que tiene el usuario para la creación de la grafica. Dicha ventana contara con tres combobox en su parte superior, que le permitira al

usuario seleccionar la escala, el tipo de grafica y el lado donde se desea graficar, tambien estara presente una tabla de 6 columnas y tantas filas como estudios tenga el archivo de estudios de pozo. La primera columna de la tabla seran celdas de tipo String que no se podran modificar, ya que tendran los nombres de los estudios, la segunda columna de la tabla seran celdas tipo booleanas con presentacion de casillas de checkbox en las que el usuario podra seleccionar los estudios a graficar. La tercera columna contendra los valores del segundo combobox de la parte superior que corresponderan al lado donde se realizaran las graficas, la cuarta columna corresponde al color de la curva a graficar y por ultimo las dos ultimas columnas seran de tipo double que recibiran los rangos de graficaci3n de las curvas.

Por ultimo la interfaz tendra un boton "Graficar", que sera activado despu3s de que todos los datos esten introducidos correctamente.

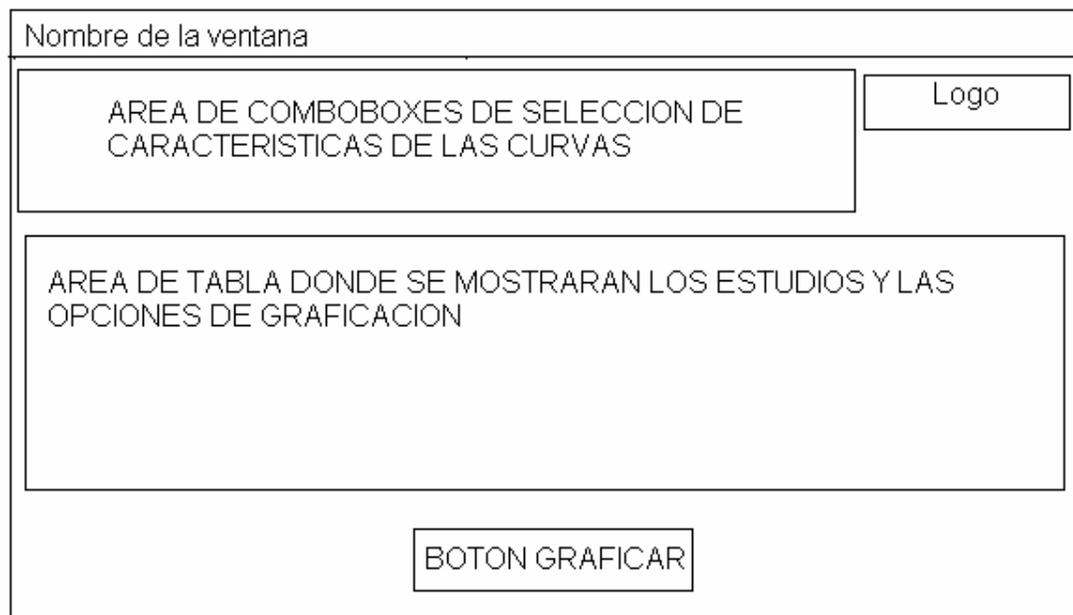


Figura 4.5 Bosquejo general de la interfaz Graficar. Fuente propia.

CAPÍTULO V

CONSTRUCCIÓN, CODIFICACIÓN Y PRUEBAS

5.1 Construcción

5.1.1 Interfaz Principal

El modelo de interfaz principal del sistema se muestra en la Figura 4.3 donde se puede apreciar que la interfaz contiene un menú de opciones de Archivo, en las cual se puede encontrar la opciones como Cargar Archivo de Datos, Cargar Archivo de Imagen, Graficar y salir. Otra opción mostrada en el menú de opciones es la pestaña de Edición, que posee Editar Archivo de Datos. En el menú de opciones también esta la pestaña ayuda que ofrece al usuario la oportunidad de tener acceso al manual de usuario de la aplicación, así como tener acceso a la versión del programa entre otros; todas estas opciones disponibles al usuario se muestran en la pantalla actual.



Figura 5.1 Interfaz Principal de Jgrafep. Fuente Propia.

5.1.2 Interfaz Abrir Archivo de Datos.

La opción Cargar Archivo de Datos se encuentra en la pestaña archivo de la ventana principal del software, al hacer clic en dicha opción se despliega una ventana correspondiente a un FileChooser, en el cual se hace un Filtrado de Archivos para que muestre por pantalla solo los archivos correspondientes a archivos de datos de perfiles de posos (Archivos LAS), como se muestra en la figura 5.2.

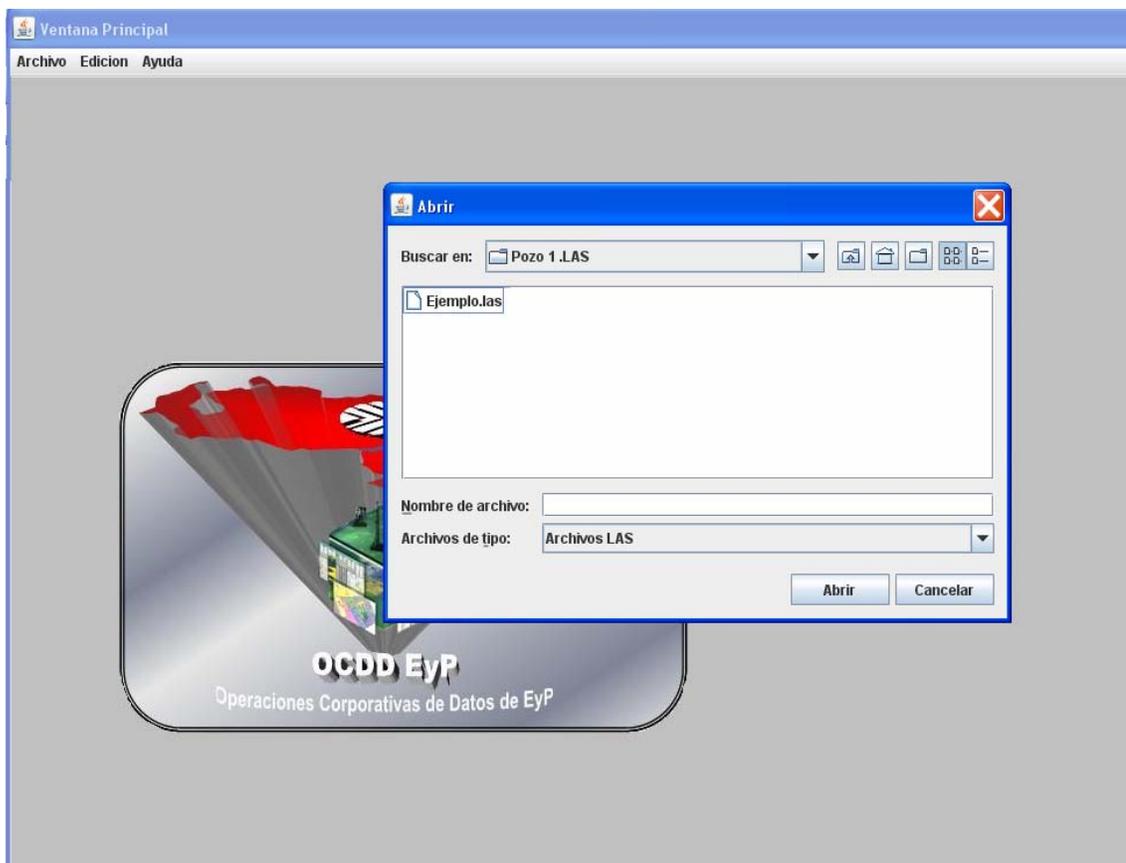


Figura 5.2 FileChooser de la Opción Cargar Archivo de Datos. Fuente Propia.

Luego de la apertura del archivo, se muestran por pantalla dos nuevas ventanas, una correspondiente al encabezado del archivo, donde se muestran 5 casillas de activación, que pertenecen a los 5 bloques existentes dentro del encabezado del

archivo. Mas abajo se muestra un campo de texto, con todas las opciones seleccionadas en las casillas de activación antes mencionadas, por defecto se ha configurado el programa para que muestre de primera instancia las opciones Versión e Información de pozo, como se muestra en la figura 5.3.

La otra ventana muestra los datos correspondiente a todos los estudios realizados, mostrados por pantalla en un tabla que tendrá tantas columnas como estudios tenga el archivo.

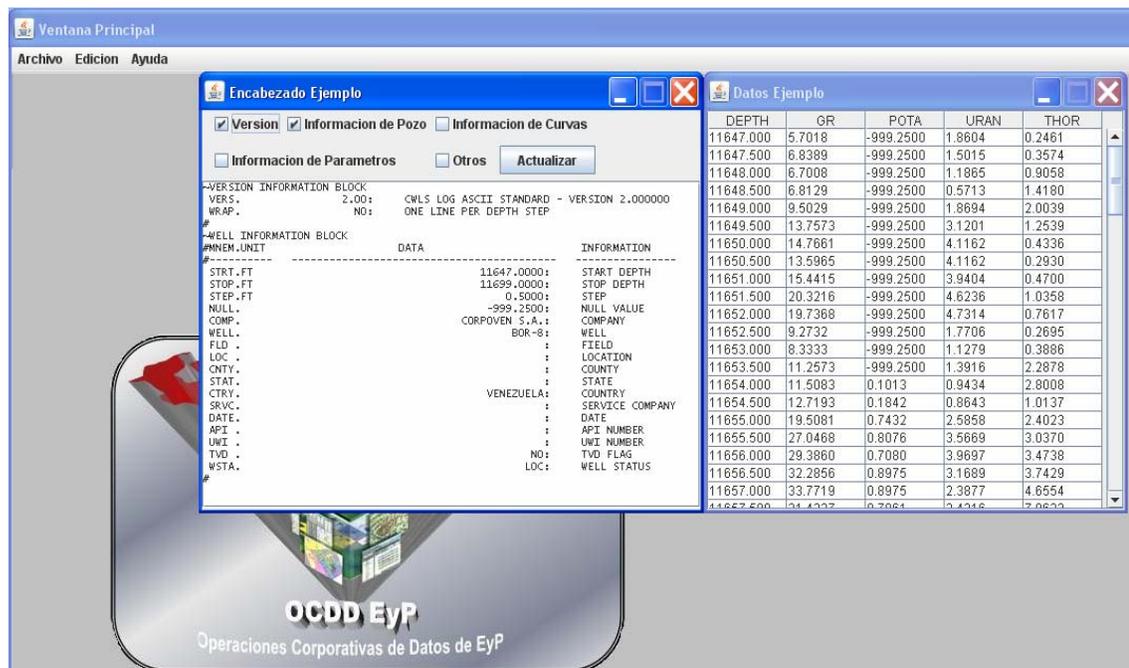


Figura 5.3 Ventanas de la Opción Cargar Archivo de Datos. Fuente Propia.

5.1.3 Interfaz Abrir Archivo de Imagen.

La opción Cargar Archivo de Imagen se encuentra en la pestaña archivo de la ventana principal del software, al hacer clic en dicha opción se despliega una ventana correspondiente a un FileChooser, en el cual se hace un Filtrado de Archivos para que

muestre por pantalla solo los archivos correspondientes a archivos de Imagen, como se muestra en la figura 5.4.

El software utiliza aplicaciones propietarias ya que la empresa posee licencias de Microsoft Office XP, 2003 y Adobe Acrobat 2007 para la apertura de los archivos de imagen debido a lo extensas de las imágenes, estas imágenes oscilan en tamaños desde 1200x6000 pixels, hasta alcanzar 1200x52000 pixels; todo esto en caso de utilizar el sistema operativo Windows, en caso contrario el programa hará la apertura de los archivos con la aplicación por defecto de el sistema GNU linux

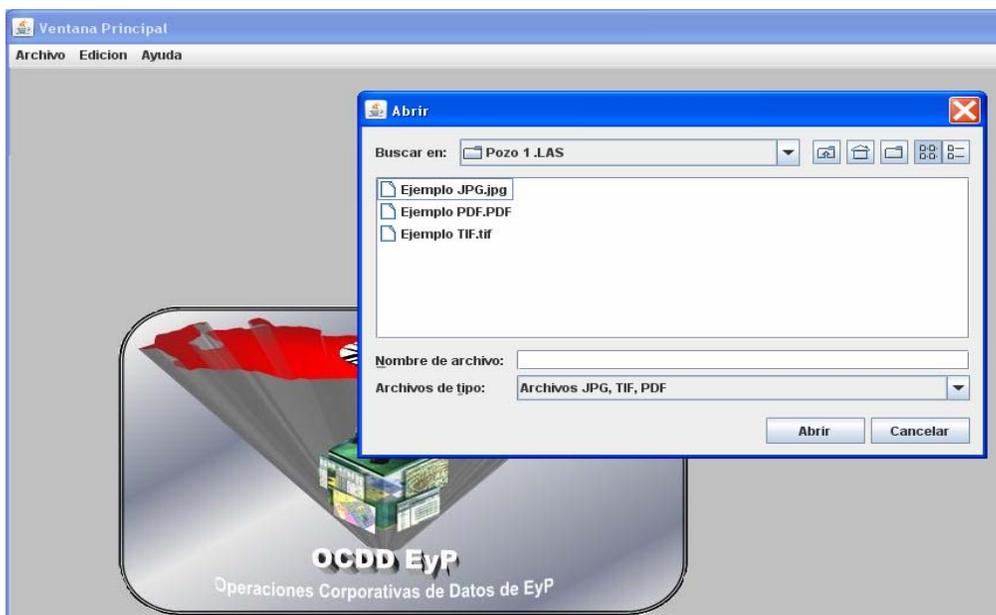


Figura 5.4 FileChooser de la Opción Cargar Archivo de Imagen. Fuente Propia.

5.1.4 Interfaz Graficar.

La opción Graficar se encuentra en la pestaña archivo de la ventana principal del software, al hacer clic en dicha opción se despliega una ventana correspondiente a un FileChooser, en el cual se hace un Filtrado de Archivos para que muestre por

pantalla solo los archivos correspondientes a archivos de datos de perfiles de pozos (Archivos LAS), como se muestra en la figura 5.5.

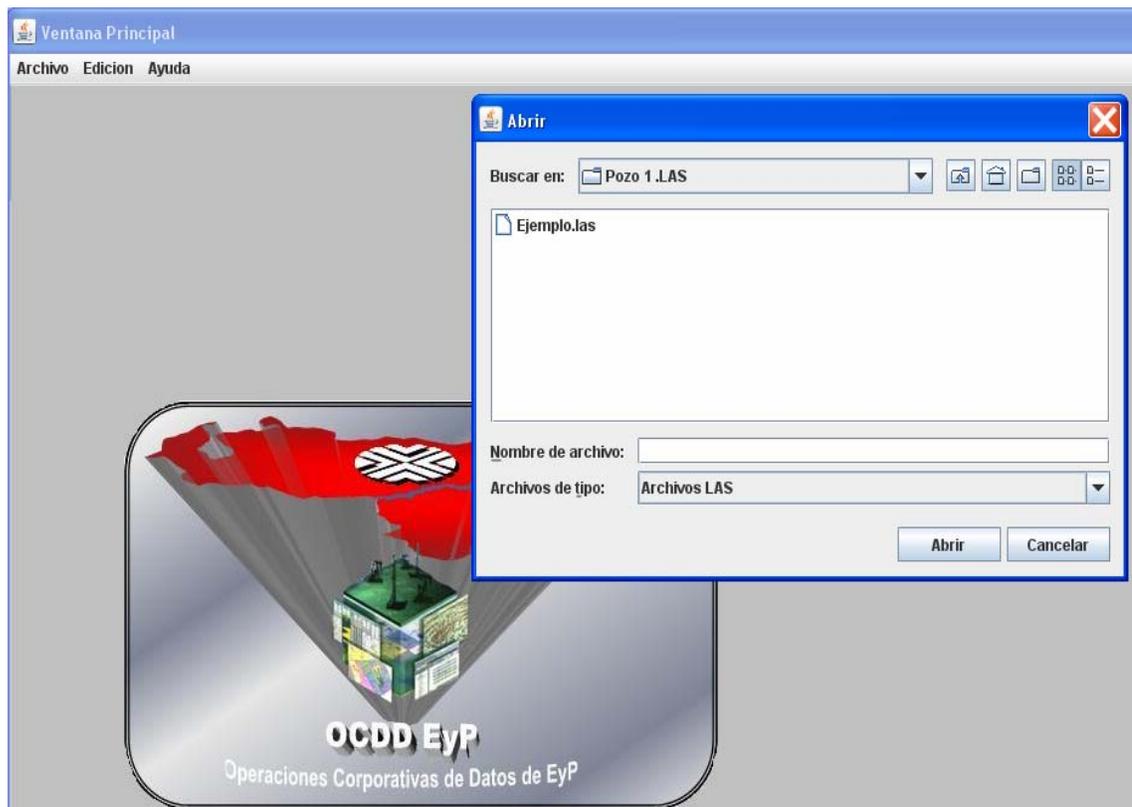


Figura 5.5 FileChooser de la Opción Graficar. Fuente Propia.

Luego de la apertura del archivo, se muestran por pantalla una nueva ventana correspondiente a las opciones que tiene el usuario para la creación de la grafica. Dicha ventana tiene los siguientes pasos a seguir:

Se selecciona la Escala en la cual se va a realizar la grafica, que consta de un combobox con las distintas Escalas Utilizadas. Figura 5.6

Se debe seleccionar el tipo de grafica, aquí se presenta un combobox con las opciones Lineal-Lineal, en caso de que las dos pistas de la grafica sean lineales, y esta

la opción Lineal-Logarítmica en el caso de que el usuario desee graficar curvas logarítmicas. Figura 5.7

El usuario debe seleccionar los estudios que se van a graficar, esto se hace en la tabla mostrada debajo de los combobox de los que se hablo anteriormente, la segunda columna de la tabla es una celda booleana que se presenta en forma de casilla de activación. Figura 5.8

Luego de haber seleccionado los estudios, se procede a escoger el lado donde se van a plasmar la grafica, se debe hacer clic en la casilla que intercepta la fila del estudio con la columna correspondiente a la columna “Graficar en Lado”, para conseguir que se active el combobox de la parte superior de nombre “Lado de la Grafica”. Figura 5.9

Ahora se procede a seleccionar el color con que se va a realizar la grafica, haciendo clic en la fila del estudio, correspondiente a la columna color, donde se desplegara un ColorChooser para escoger el color. Figura 5.10

Se introducen los rangos derechos e izquierdos, haciendo clic en las casillas correspondientes y escribiéndolos. Figura 5.11.

Se hace clic en el botón Graficar.

En la labor de diseño, se obtuvo una visión precisa y clara de la forma de trabajo del usuario. Se logró hacer un diseño efectivo de los objetos que conformaran las ventanas y se logró crear un diseño de interfaz amigable y práctico para los usuarios.

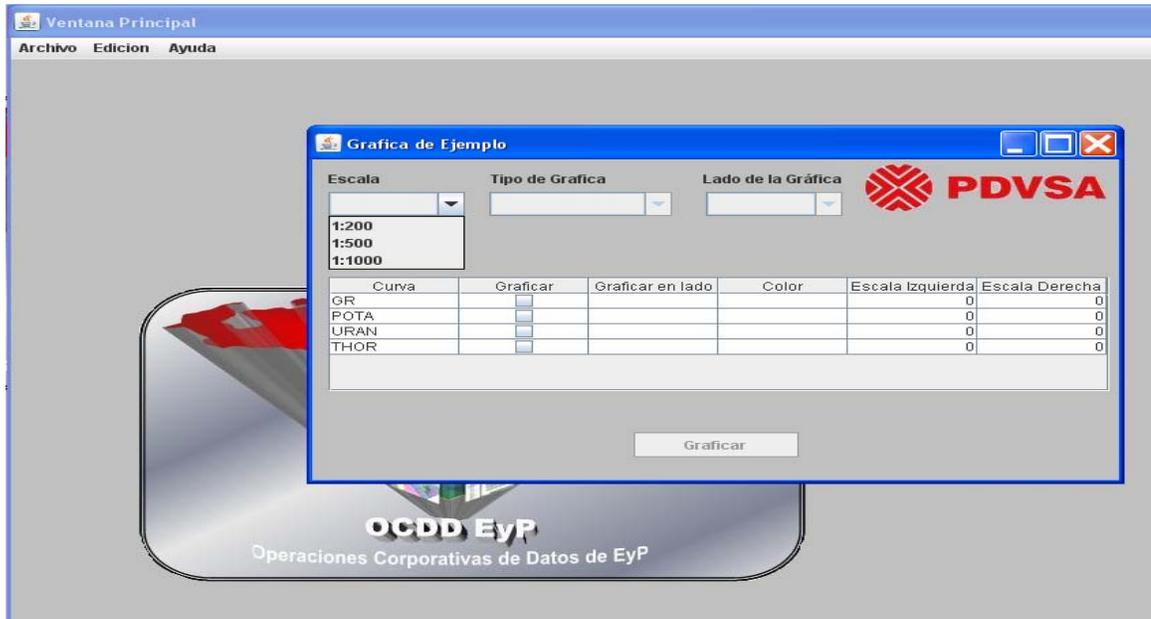


Figura 5.6 ComboBox Escala. Fuente Propia.

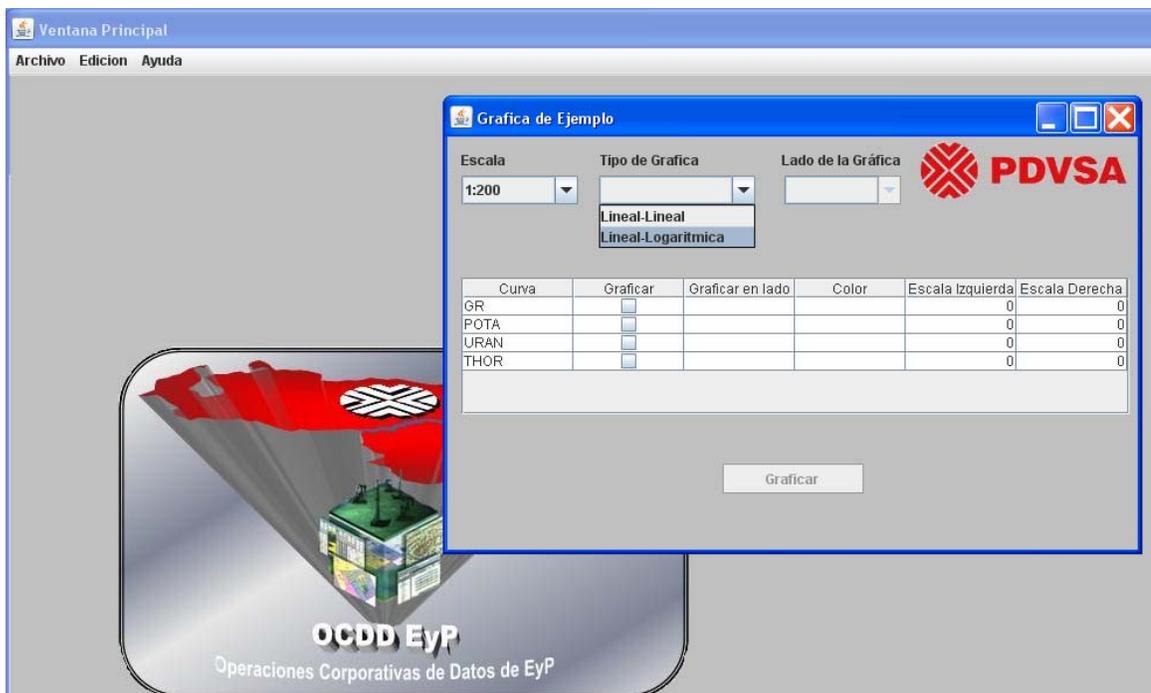


Figura 5.7 ComboBox Tipo de Grafica. Fuente Propia.

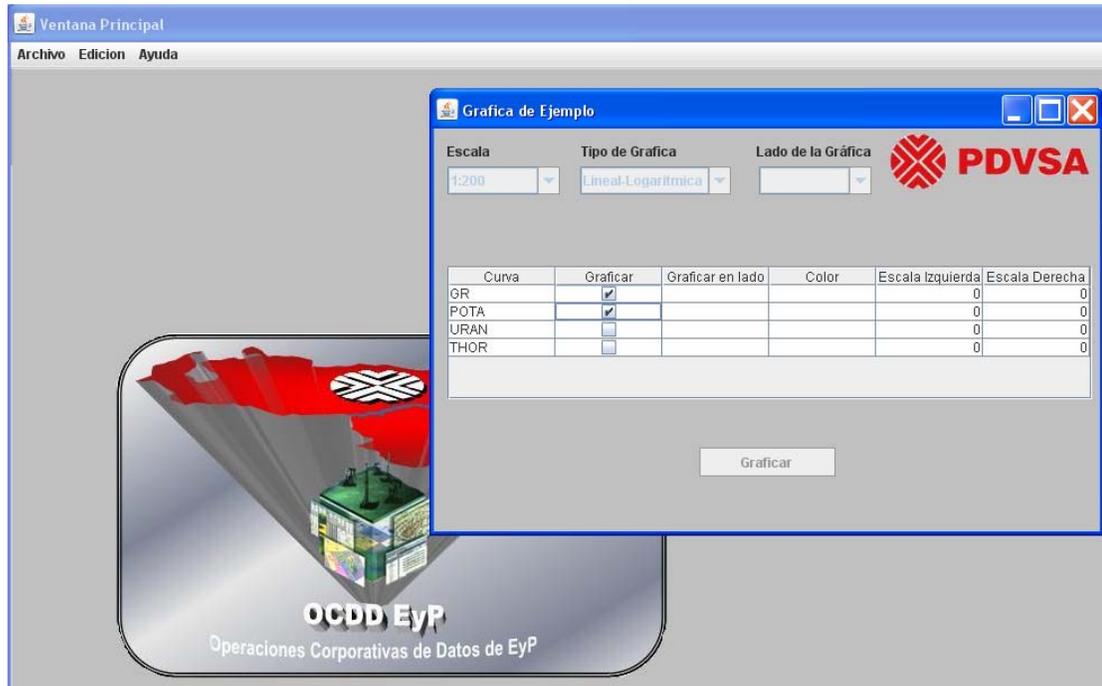


Figura 5.8 Celda booleanas de Selección de curvas. Fuente Propia.

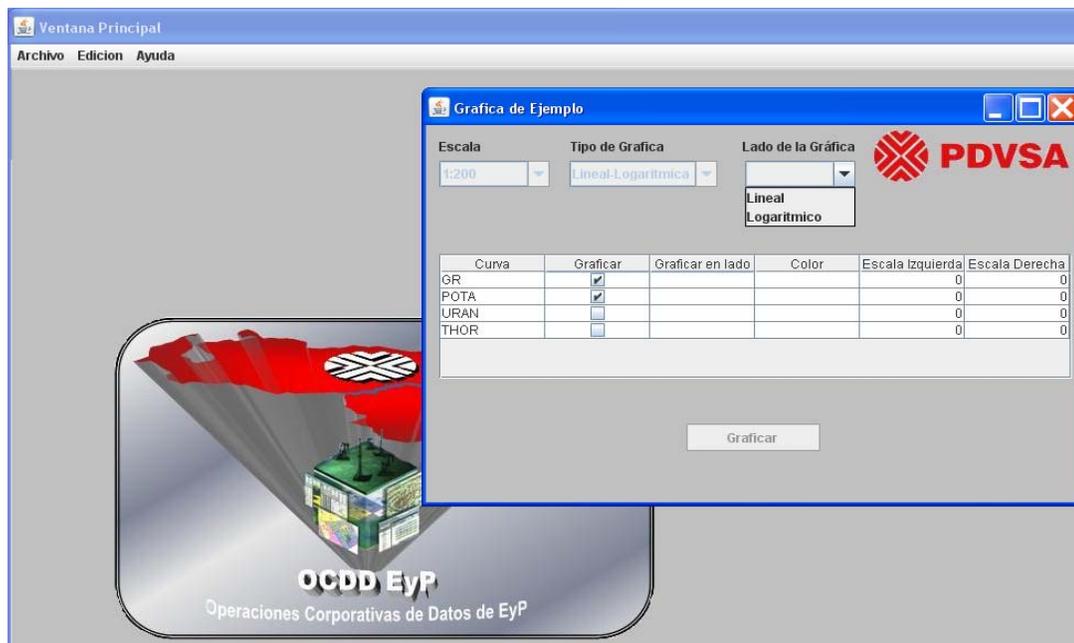


Figura 5.9 ComboBox Lado de la Grafica. Fuente Propia.

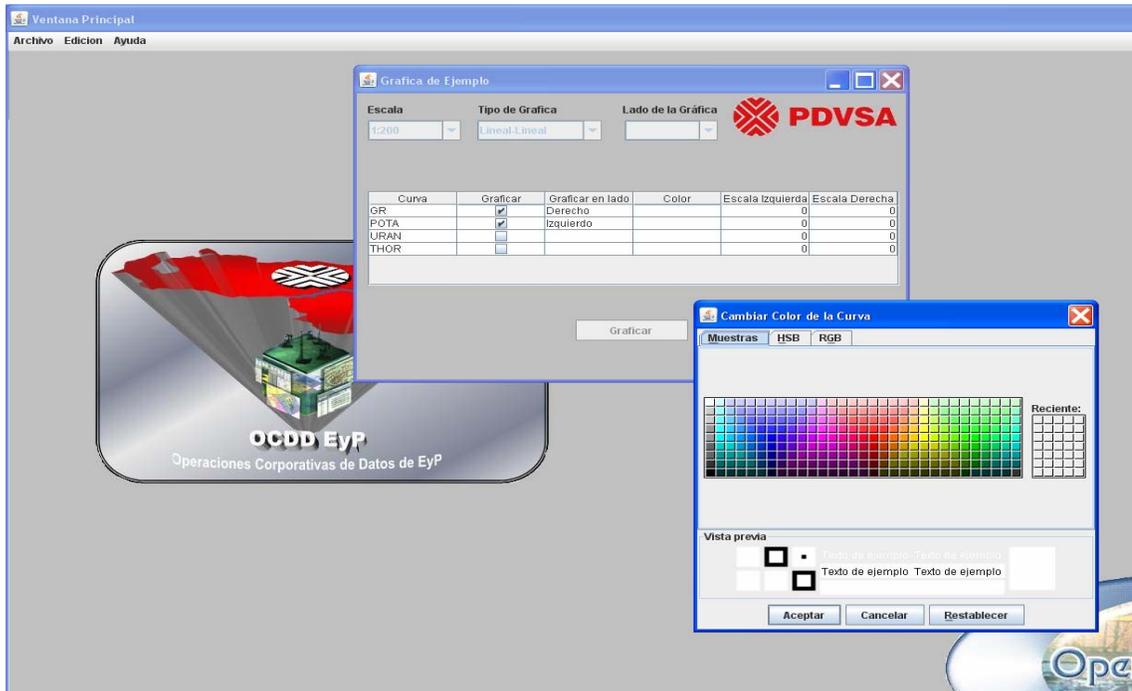


Figura 5.10 ColorChooser. Fuente Propia.

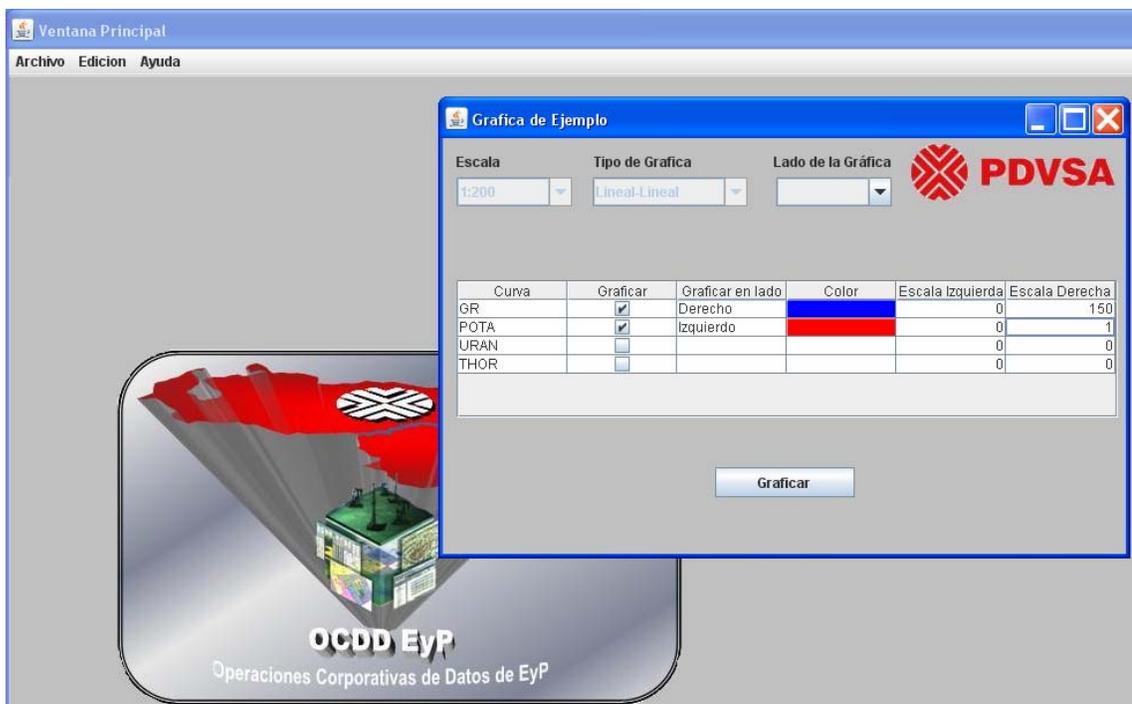


Figura 5.11 Rangos de Graficación. Fuente Propia.

5.2 Codificación

Se implementan los códigos fuentes y ejecutables correspondientes a los casos de uso Mostrar Registro Datos, Mostrar Registro Imagen, Graficar Curvas, Editar Archivo Datos, con el propósito de obtener la versión beta de JGrafep.

5.2.1 Implementación del caso de uso Principal

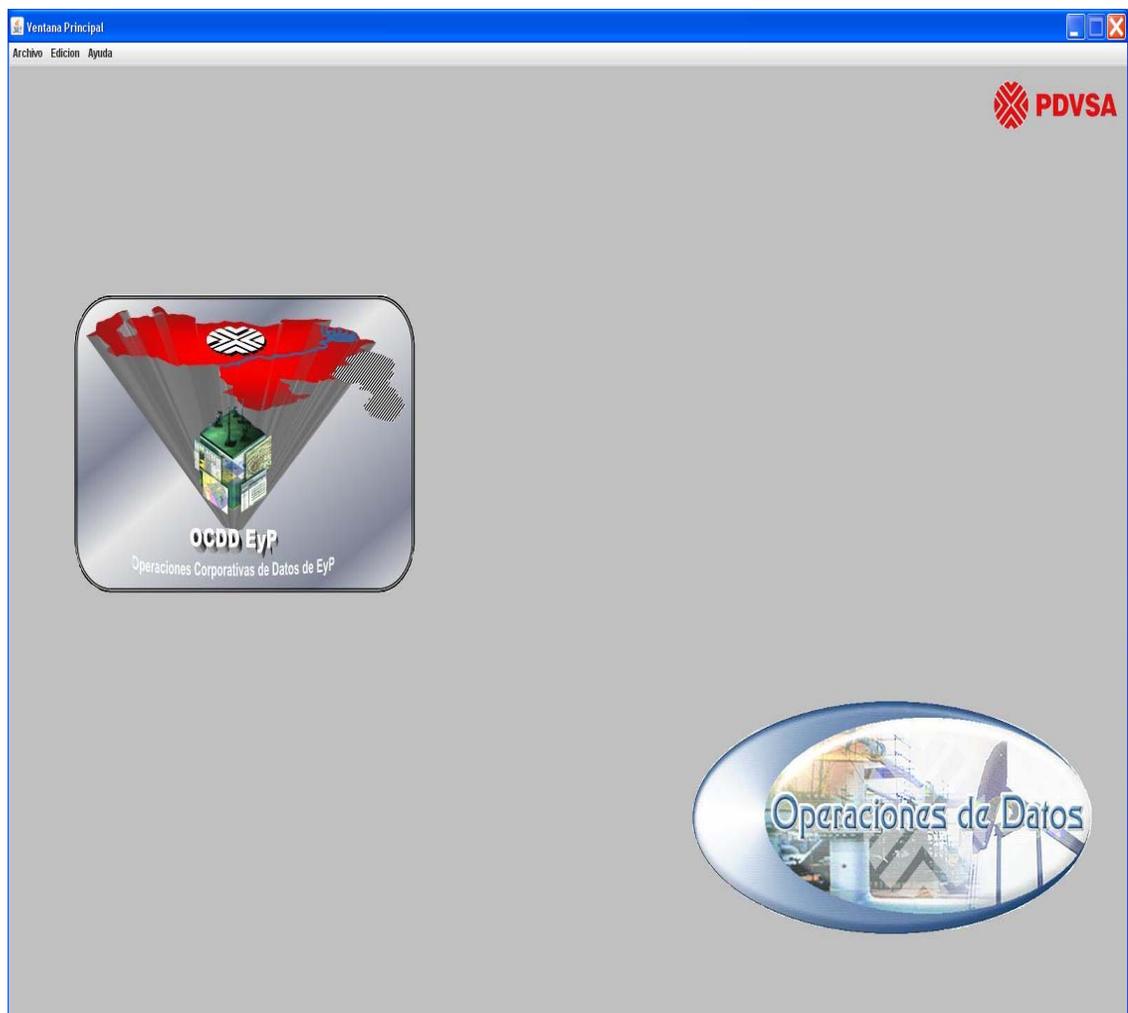


Figura 5.12 Ventana Principal. Fuente Propia.

Nombre del Archivo: Principal.java

```

/*
 * Principal.java
 *
 * Created on 8 de febrero de 2008, 17:42
 */
package graficador;
import java.io.IOException;
/**
 *
 * @author Julio Martinez
 */
public class Principal extends javax.swing.JFrame
{
    java.io.File NombreArchivo,ArchivoImagen;
    String nomarc,direccion;
    java.io.BufferedReader streami;
    Process proceso;
    boolean PDFS=false,JPG=false;
    InformacionCurvas Datos;
    Filtro filtro, filtro2;
    /** Creates new form Principal */
    public Principal() {
        initComponents();
        //Se almacena en una lista doble todos los estudios
        Datos = new InformacionCurvas();
        filtro = new Filtro();
        /* extensiones que se quieren aÃ±adir*/
        filtro.addExtension("las");
        /* descripcion del filtro*/
        filtro.setDescription("Archivos LAS");
        ArchivoLas.addChoosableFileFilter(filtro);
        filtro2 = new Filtro();
    }
}

```

```

        /* extensiones que quieres añadir */
        filtro2.addExtension("Jpg");
        filtro2.addExtension("Tif");
        filtro2.addExtension("Jpeg");
        filtro2.addExtension("Tiff");
        filtro2.addExtension("JPG");
        filtro2.addExtension("TIF");
        filtro2.addExtension("TIFF");
        filtro2.addExtension("Pdf");
        /* descripcion del filtro */
        filtro2.setDescription("Archivos JPG, TIF, PDF");
        // Aqui se agrega el filtro creado con las extensiones
al FileChooser
        ArchivoImg.addChoosableFileFilter(filtro2);
    }
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        ArchivoLas = new javax.swing.JFileChooser();
        ArchivoImg = new javax.swing.JFileChooser();
        Fondo = new javax.swing.JPanel();
        LogoOCDD = new javax.swing.JLabel();
        LogoPDVSA = new javax.swing.JLabel();
        LogoOC = new javax.swing.JLabel();
        MenuBar = new javax.swing.JMenuBar();
        Menu = new javax.swing.JMenu();
        CargarArchivoDatos = new javax.swing.JMenuItem();
        CargarArchivoIMagen = new javax.swing.JMenuItem();
        Graficar = new javax.swing.JMenuItem();
        SalirMenu = new javax.swing.JMenuItem();
        jMenu1 = new javax.swing.JMenu();
        Editar = new javax.swing.JMenuItem();
        Ayuda = new javax.swing.JMenu();
        Manual = new javax.swing.JMenuItem();

```

```

Acerca = new javax.swing.JMenuItem();

ArchivoLas.setAcceptAllFileFilterUsed(false);

ArchivoImg.setAcceptAllFileFilterUsed(false);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Ventana Principal");
    setBackground(new java.awt.Color(255, 255, 255));
    setBounds(new java.awt.Rectangle(0, 0, 1280, 1024));
    setMaximizedBounds(new java.awt.Rectangle(0, 0, 1280,
1024));

    setMinimumSize(new java.awt.Dimension(1024, 768));
    setName("VentanaPrincipal"); // NOI18N
    setResizable(false);
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(java.awt.event.WindowEvent
evt) {
            formWindowClosing(evt);
        }
    });

Fondo.setBackground(java.awt.Color.lightGray);
Fondo.setPreferredSize(new java.awt.Dimension(1024,
768));

LogoOCDD.setIcon(new
javax.swing.ImageIcon("C:\\Documents and Settings\\Julio
C\\Escritorio\\JGrafep\\Imagen\\OCDD.jpg")); // NOI18N
LogoOCDD.setOpaque(true);

LogoPDVSA.setIcon(new
javax.swing.ImageIcon("C:\\Documents and Settings\\Julio
C\\Escritorio\\JGrafep\\Imagen\\PDVSA.jpg")); // NOI18N

```

```

        LogoOC.setIcon(new javax.swing.ImageIcon("C:\\Documents
and Settings\\Julio C\\Escritorio\\JGrafep\\Imagen\\ODD.jpg")); //
NOI18N

```

```

        javax.swing.GroupLayout FondoLayout = new
javax.swing.GroupLayout(Fondo);
        Fondo.setLayout(FondoLayout);
        FondoLayout.setHorizontalGroup(

```

```

FondoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
FondoLayout.createSequentialGroup()

```

```

        .addContainerGap(1846, Short.MAX_VALUE)

```

```

        .addComponent(LogoPDVSA)

```

```

        .addContainerGap()

```

```

        .addGroup(FondoLayout.createSequentialGroup()

```

```

        .addGap(88, 88, 88)

```

```

        .addComponent(LogoOCDD)

```

```

        .addContainerGap(1465, Short.MAX_VALUE))

```

```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
FondoLayout.createSequentialGroup()

```

```

        .addContainerGap(1386, Short.MAX_VALUE)

```

```

        .addComponent(LogoOC)

```

```

        .addGap(82, 82, 82))

```

```

);

```

```

FondoLayout.setVerticalGroup(

```

```

FondoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(FondoLayout.createSequentialGroup()

```

```

        .addContainerGap()

```

```

        .addComponent(LogoPDVSA)
        .addGap(108, 108, 108)
        .addComponent(LogoOCDD)
        .addGap(124, 124, 124)
        .addComponent(LogoOC)
        .addGap(136, 136, 136))
    );

    Menu.setText("Archivo");

CargarArchivoDatos.setAccelerator(javax.swing.KeyStroke.getKeyStroke
(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));
    CargarArchivoDatos.setText("Cargar Archivo de Datos");
    CargarArchivoDatos.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            CargarArchivoDatosActionPerformed(evt);
        }
    });
    Menu.add(CargarArchivoDatos);

CargarArchivoIMagen.setAccelerator(javax.swing.KeyStroke.getKeyStrok
e(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));
    CargarArchivoIMagen.setText("Cargar Archivo de Imagen");
    CargarArchivoIMagen.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            CargarArchivoIMagenActionPerformed(evt);
        }
    });

```

```

    });
    Menu.add(CargarArchivoIMagen);

Graficar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.
event.KeyEvent.VK_G, java.awt.event.InputEvent.CTRL_MASK));
    Graficar.setText("Graficar");
    Graficar.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            GraficarActionPerformed(evt);
        }
    });
    Menu.add(Graficar);

SalirMenu.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt
.event.KeyEvent.VK_X, java.awt.event.InputEvent.CTRL_MASK));
    SalirMenu.setText("Salir");
    SalirMenu.addMouseListener(new
java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent
evt) {
            SalirMenuMouseClicked(evt);
        }
    });
    SalirMenu.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            SalirMenuActionPerformed(evt);
        }
    });
    Menu.add(SalirMenu);

```

```

MenuBar.add(Menu);
jMenu1.setText("Edicion");

Editar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.ev
ent.KeyEvent.VK_E, java.awt.event.InputEvent.CTRL_MASK));
    Editar.setText("Editar Archivo de Datos");
    Editar.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            EditarActionPerformed(evt);
        }
    });
jMenu1.add(Editar);

MenuBar.add(jMenu1);

Ayuda.setText("Ayuda");

Manual.setText("Manual de Usuario");
Manual.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        ManualActionPerformed(evt);
    }
});
Ayuda.add(Manual);

Acerca.setText("Acerca de...");
Acerca.addActionListener(new
java.awt.event.ActionListener() {

```

```

        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            AcercaActionPerformed(evt);
        }
    });
    Ayuda.add(Acerca);

    MenuBar.add(Ayuda);

    setJMenuBar(MenuBar);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent(Fondo,
javax.swing.GroupLayout.DEFAULT_SIZE, 2042, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent(Fondo,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 985, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

@SuppressWarnings("static-access")

```

```

private void
CargarArchivoDatosActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_CargarArchivoDatosActionPerformed
    //Función para cargar el archivo de datos con las curvas
    try
    {
        int returnVal = ArchivoLas.showOpenDialog(this);
        if (returnVal == ArchivoLas.APPROVE_OPTION)
        {
            NombreArchivo =
ArchivoLas.getSelectedFile();

            nomarc=NombreArchivo.getName();
            nomarc=nomarc.replace(".las","");
            nomarc=nomarc.replace("-","");
            Datos Dat = new Datos();
            Dat.setTitle("Datos "+nomarc);
            System.gc(); System.gc();
            InformacionEncabezado encabezado=new
InformacionEncabezado();
            encabezado.Recibir_Archivo(NombreArchivo);
            Encabezado Enc = new Encabezado(encabezado);
            Enc.setTitle("Encabezado "+nomarc);
            Enc.setBounds(270, 58, 480, 400);
            System.gc();
            Datos.AsignarNulo(encabezado.ObtenerNulo());
            if(Datos.Recibir_Archivo(NombreArchivo))
            {
                System.gc(); System.gc();

            Dat.MostrarDatos(Datos.ObtenerCoordenadas(),
Datos.ObtenerTitulos());

                Dat.setBounds(750, 58,
Dat.ObtenerAncho()+28, 400);

                Enc.setVisible(true);
                Dat.setVisible(true);
            }
        }
    }
}

```

```

        }
    }
}
catch (Exception e)
{

javax.swing.JOptionPane.showMessageDialog(null,"No se pudo abrir el
archivo"+e.getMessage(),"Seleccion de Archivos",3);

}
} //GEN-LAST:event_CargarArchivoDatosActionPerformed

private void SalirMenuMouseClicked(java.awt.event.MouseEvent
evt) { //GEN-FIRST:event_SalirMenuMouseClicked
    if(PDFS||JPG)
    {
        matarProceso();
    }
    System.exit(0);

} //GEN-LAST:event_SalirMenuMouseClicked

private void SalirMenuActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_SalirMenuActionPerformed
    // TODO add your handling code here:
    if(PDFS||JPG)
    {
        matarProceso();
    }
    System.exit(0);

} //GEN-LAST:event_SalirMenuActionPerformed

private void matarProceso()

```

```
{
    String osName = System.getProperty("os.name");
    String cmd = "";
    String cmd2 = "";
    if(osName.toUpperCase().contains("WIN"))
    { //S.O. Windows
        if(PDFS)
        {
            cmd="tskill Acrobat";
        }
        if(JPG)
        {
            cmd2="tskill OIS";
        }
    }
    else
    { //Solo ha sido probado en win y linux
        if(PDFS)
        {
            cmd="killall Acrobat";
        }
        if(JPG)
        {
            cmd2="killall openoffice";
        }
    }
    Process hijo;
    if(PDFS)
    {
        try
        {
            hijo = Runtime.getRuntime().exec(cmd);
            hijo.waitFor();
            if ( hijo.exitValue()==0)

```

```

        {
            System.out.println("Acrobat    acabado    con
exito");
        }
        else
        {
            System.out.println("Incapaz    de    matar
Acrobat. Exit code: " + hijo.exitValue()+"n");
        }
    }
    catch (IOException e)
    {

        javax.swing.JOptionPane.showMessageDialog(null,"Imposible    matar    el
proceso "+e.getMessage(),"Cerrando Aplicacion",1);
    }
    catch (InterruptedException e)
    {

        javax.swing.JOptionPane.showMessageDialog(null,"Imposible    matar    el
proceso "+e.getMessage(),"Cerrando Aplicacion",1);
    }
    PDFS=false;
}
if(JPG)
{
    try
    {
        hijo = Runtime.getRuntime().exec(cmd2);
        hijo.waitFor();
        if ( hijo.exitValue()==0)
        {
            System.out.println("Office    matado    con
exito");
        }
    }
}

```

```

        else
        {
            System.out.println("Incapaz de matar Office.
Exit code: " + hijo.exitValue()+"n");
        }
    }
    catch (IOException e)
    {

        javax.swing.JOptionPane.showMessageDialog(null,"Imposible matar el
proceso "+e.getMessage(),"Cerrando Aplicacion",1);
    }
    catch (InterruptedException e)
    {

        javax.swing.JOptionPane.showMessageDialog(null,"Imposible matar el
proceso "+e.getMessage(),"Cerrando Aplicacion",1);
    }
    }
    JPG=false;
}

}

@SuppressWarnings("static-access")
private void
CargarArchivoIMagenActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_CargarArchivoIMagenActionPerformed
    // Aqui se le agregan las extensiones de los archivos de
imagen a cargar
    try
    {
        // Se abre la ventana del FileChooser
        int returnVal = ArchivoImg.showOpenDialog(this);
        if (returnVal == ArchivoImg.APPROVE_OPTION)
        {

```

```
        // Se selecciona el archivo de imagen
        ArchivoImagen = ArchivoImg.getSelectedFile();
        // Se guarda el nombre del archivo
        nomarc=ArchivoImagen.getName();
        nomarc=nomarc.toUpperCase();

        if(nomarc.contains(".JPG") || nomarc.contains(".JPEG"))
        {
            JPG=true;
            ImagenJPG jpg = new ImagenJPG();

            jpg.PRUEBA2JPG(ArchivoImagen.getAbsolutePath());
        }

        if(nomarc.contains(".TIF") || nomarc.contains(".TIFF"))
        {
            JPG=true;
            ImagenTIF TIF = new ImagenTIF();

            TIF.ImagenTIF(ArchivoImagen.getAbsolutePath());
        }
        if(nomarc.contains(".PDF"))
        {
            PDFS=true;
            ImagenPDF PDF = new ImagenPDF();

            PDF.ImagenPDF(ArchivoImagen.getAbsolutePath());
        }
    }
}

catch (Exception e)
{
```

```

        javax.swing.JOptionPane.showMessageDialog(null,"No se
pudo abrir el archivo "+e.getMessage(),"Seleccion de Archivos",3);
    }

} //GEN-LAST:event_CargarArchivoIMagenActionPerformed

@SuppressWarnings("static-access")
private void
GraficarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_GraficarActionPerformed
    boolean ban=false;
    int returnVal = ArchivoLas.showOpenDialog(this);
    if (returnVal == ArchivoLas.APPROVE_OPTION)
    {
        Graficar graf = new Graficar();
        graf.mostrarcurvas(ArchivoLas.getSelectedFile());
        ban=graf.SioNo();
        if(ban)
        {
            graf.dispose();
        }
        else
        {
            graf.setBounds(150, 458, 600, 400);
            graf.setVisible(true);
        }
    }
} //GEN-LAST:event_GraficarActionPerformed

private void
EditarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_EditarActionPerformed
    // TODO add your handling code here:
    Editar Edit = new Editar();
    Edit.Editar();
}

```

```

        Edit.setVisible(true);
    }//GEN-LAST:event_EditActionPerformed

    private void formWindowClosing(java.awt.event.WindowEvent
    evt) { //GEN-FIRST:event_formWindowClosing
        if(PDFS || JPG)
        {
            matarProceso();
        }
        System.exit(0);
    } //GEN-LAST:event_formWindowClosing

    private void
    ManualActionPerformed(java.awt.event.ActionEvent    evt)    { //GEN-
    FIRST:event_ManualActionPerformed

        PDFS=true;
        ImagenPDF PDF = new ImagenPDF();
        PDF.ImagenPDF("Manual de Usuario.pdf");

    } //GEN-LAST:event_ManualActionPerformed

    private void
    AcercaActionPerformed(java.awt.event.ActionEvent    evt)    { //GEN-
    FIRST:event_AcercaActionPerformed

    javax.swing.JOptionPane.showMessageDialog(null, "JGrafep\n" +
    "Creado
    por: Julio C. Martinez G.\n"+
    "Version
    1.0"

    , "Ayuda", 3);
    } //GEN-LAST:event_AcercaActionPerformed
    /**

```

```

    * @param args the command line arguments
    */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new Principal().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-
BEGIN:variables
    private javax.swing.JMenuItem Acerca;
    private javax.swing.JFileChooser ArchivoImg;
    private javax.swing.JFileChooser ArchivoLas;
    private javax.swing.JMenu Ayuda;
    private javax.swing.JMenuItem CargarArchivoDatos;
    private javax.swing.JMenuItem CargarArchivoIMagen;
    private javax.swing.JMenuItem Editar;
    private javax.swing.JPanel Fondo;
    private javax.swing.JMenuItem Graficar;
    private javax.swing.JLabel LogoOC;
    private javax.swing.JLabel LogoOCDD;
    private javax.swing.JLabel LogoPDVSA;
    private javax.swing.JMenuItem Manual;
    private javax.swing.JMenu Menu;
    private javax.swing.JMenuBar MenuBar;
    private javax.swing.JMenuItem SalirMenu;
    private javax.swing.JMenu jMenu1;
    // End of variables declaration//GEN-END:variables
}

```

5.2.2 Implementación del caso de uso Abrir Archivo de Datos

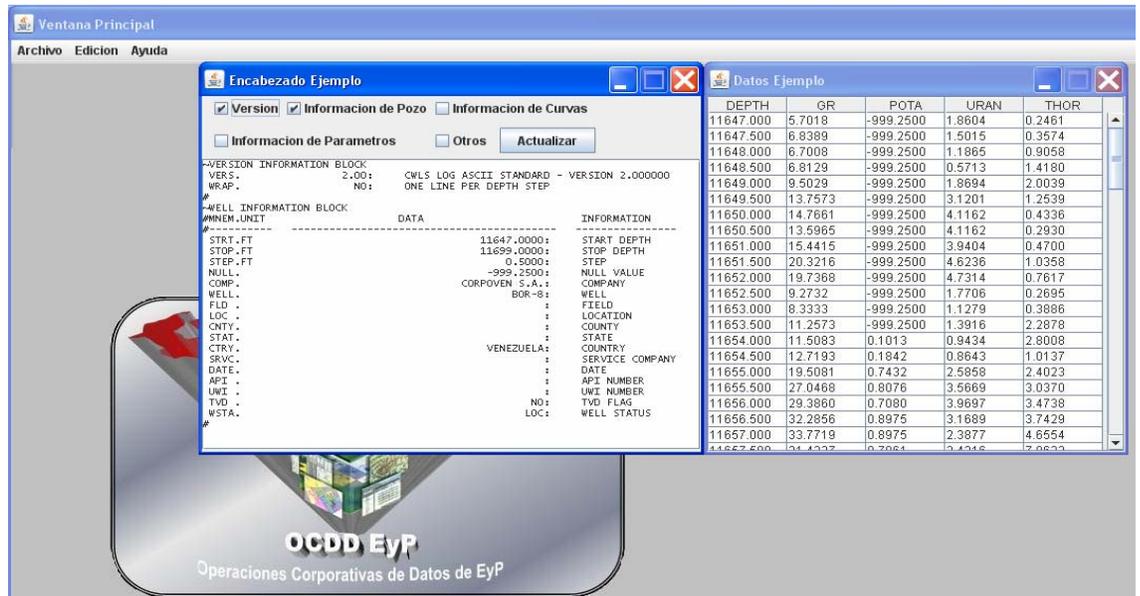


Figura 5.13 Ventanas con los datos del archivo. Fuente Propia.

Nombre del Archivo: Encabezado.java

```

/*
 * Encabezado.java
 *
 * Created on 19 de diciembre de 2008, 03:47 PM
 */

package graficador;

import java.io.File;

/**Esta clase se encarga de mostrar el encabezado del archivo en
una ventana
 *
 * @author Julio C

```

```

*/

public class Encabezado extends javax.swing.JFrame {
    /**Almacen la informaci3n del encabezado.*/
    InformacionEncabezado encabezado;
    /** Constructor de la clase Encabezado */
    public Encabezado(InformacionEncabezado encab)
    {
        initComponents();
        encabezado=new InformacionEncabezado();
        encabezado=encab;
        jTextArea1.setText("");
        jTextArea1.append(encabezado.ObtenerEncabezado());
    }
    /** This method is called from within the constructor to
    * initialize the form.
    * WARNING: Do NOT modify this code. The content of this
method is
    * always regenerated by the Form Editor.
    */
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        Version = new javax.swing.JCheckBox();
        Pozo = new javax.swing.JCheckBox();
        InfoCurvas = new javax.swing.JCheckBox();
        Parametros = new javax.swing.JCheckBox();
        Otros = new javax.swing.JCheckBox();
        Actualizar = new javax.swing.JButton();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("Encabezado");
setBounds(new java.awt.Rectangle(900, 50, 0, 0));
setName("Encabezado"); // NOI18N
setResizable(false);
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent
evt) {
        Cerrando(evt);
    }
});

jTextArea1.setColumns(1);
jTextArea1.setEditable(false);
jTextArea1.setFont(new java.awt.Font("Lucida Console",
0, 10));

jTextArea1.setRows(5);
jTextArea1.setDoubleBuffered(true);
jTextArea1.setFocusCycleRoot(true);
jTextArea1.setFocusTraversalPolicyProvider(true);
jScrollPane.setViewportView(jTextArea1);

Version.setSelected(true);
Version.setText("Version");

Pozo.setSelected(true);
Pozo.setText("Informacion de Pozo");

InfoCurvas.setText("Informacion de Curvas");

Parametros.setText("Informacion de Parametros");

Otros.setText("Otros");

```

```

        Actualizar.setText("Actualizar");
        Actualizar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                ActualizarActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(Pozo)
                            .addComponent(Parametros))
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(Otros)

```

```

        .addGap(8, 8, 8)
        .addComponent(Actualizar))
        .addComponent(InfoCurvas)))
        .addComponent(jScrollPane,
javax.swing.GroupLayout.DEFAULT_SIZE, 381, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)

        .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)

        .addComponent(Version)
        .addComponent(Pozo)
        .addComponent(InfoCurvas))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)

        .addComponent(Parametros)
        .addComponent(Otros)
        .addComponent(Actualizar))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jScrollPane,
javax.swing.GroupLayout.PREFERRED_SIZE, 296,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(57, 57, 57))
    );

```

```

        pack();
    }// </editor-fold>//GEN-END:initComponents
    /**Funci3n para actualizar la ventana del encabezado*/
    private void
ActualizarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_ActualizarActionPerformed
        JTextAreal.setText("");
        encabezado.Mostrar_Encabezado(Version.isSelected(),
Pozo.isSelected(), InfoCurvas.isSelected(), Parametros.isSelected(),
Otros.isSelected());
        JTextAreal.append(encabezado.ObtenerEncabezado());
    } //GEN-LAST:event_ActualizarActionPerformed

    private void Cerrando(java.awt.event.WindowEvent evt)
{ //GEN-FIRST:event_Cerrando
        //Liberando memoria de la maquina virtual
        System.gc();
        System.gc();
        System.gc();
        System.gc();
    } //GEN-LAST:event_Cerrando

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                // new Encabezado().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-
BEGIN:variables

```

```
private javax.swing.JButton Actualizar;  
private javax.swing.JCheckBox InfoCurvas;  
private javax.swing.JCheckBox Otros;  
private javax.swing.JCheckBox Parametros;  
private javax.swing.JCheckBox Pozo;  
private javax.swing.JCheckBox Version;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTextArea jTextArea1;  
// End of variables declaration//GEN-END:variables  
  
}
```

Nombre del Archivo: **Datos.java**

```
/*  
 * Datos.java  
 *  
 * Created on 29/01/2009, 09:56:15 AM  
 */  
package graficador;  
  
import java.awt.Color;  
  
/**  
 *  
 * @author Julio Martinez  
 */  
public class Datos extends javax.swing.JFrame {  
    /** Creates new form Datos */  
    int ancho=0;  
    int fila,columna;  
    public Datos() {  
        initComponents();  
    }  
}
```

```

    }
    public void MostrarDatos(Object temp[][], Object Tit[])
    {
        int anchocel=0,cantcol=0;
        Datos.setModel(new
javax.swing.table.DefaultTableModel(temp,Tit));
        anchocel=Datos.getWidth();
        cantcol= Datos.getColumnCount();
        if(cantcol<6)
        {
            ancho=anchocel*cantcol;
        }
        else
        {
            ancho=anchocel*6;
        }
    }
    @SuppressWarnings("static-access")
    public int ObtenerAncho()
    {
        return ancho;
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        jScrollPane2 = new javax.swing.JScrollPane();
        Datos = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOS
E);

        setTitle("Datos");
        setBounds(new java.awt.Rectangle(950, 500, 0, 0));

```

```

setResizable(false);
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent
evt) {
        Cerrando(evt);
    }
});

jScrollPane2.setAutoscrolls(true);

Datos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null}
    },
    new String [] {
        "Title 1"
    }
));

Datos.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
Datos.setCellSelectionEnabled(true);
Datos.setDoubleBuffered(true);
Datos.addMouseListener(new java.awt.event.MouseAdapter()
{
    public void mouseClicked(java.awt.event.MouseEvent
evt) {
        DatosMouseClicked(evt);
    }
});
jScrollPane2.setViewportViewView(Datos);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

```

```

layout.createParallelGroup( javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent( jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 319, Short.MAX_VALUE)
        );
layout.setVerticalGroup(

layout.createParallelGroup( javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent( jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 300, Short.MAX_VALUE)
        );

pack();
} // </editor-fold> // GEN-END: initComponents

private void Cerrando( java.awt.event.WindowEvent evt )
{ // GEN-FIRST: event_Cerrando
    // Liberando memoria de la maquina virtual
    System.gc();
    System.gc();
    System.gc();
    System.gc();
} // GEN-LAST: event_Cerrando

private void DatosMouseClicked( java.awt.event.MouseEvent
evt ) { // GEN-FIRST: event_DatosMouseClicked
    Datos.setRowSelectionAllowed( true );
    Datos.setColumnSelectionAllowed( false );
    fila = Datos.getSelectedRow();

    Datos.setSelectionBackground( Color.YELLOW );
} // GEN-LAST: event_DatosMouseClicked

/**

```

```

* @param args the command line arguments
*/
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Datos().setVisible(true);
        }
    });
}
// Variables declaration - do not modify//GEN-
BEGIN:variables
private javax.swing.JTable Datos;
private javax.swing.JScrollPane jScrollPane2;
// End of variables declaration//GEN-END:variables
}

```

5.2.3 Implementación del caso de uso Graficar

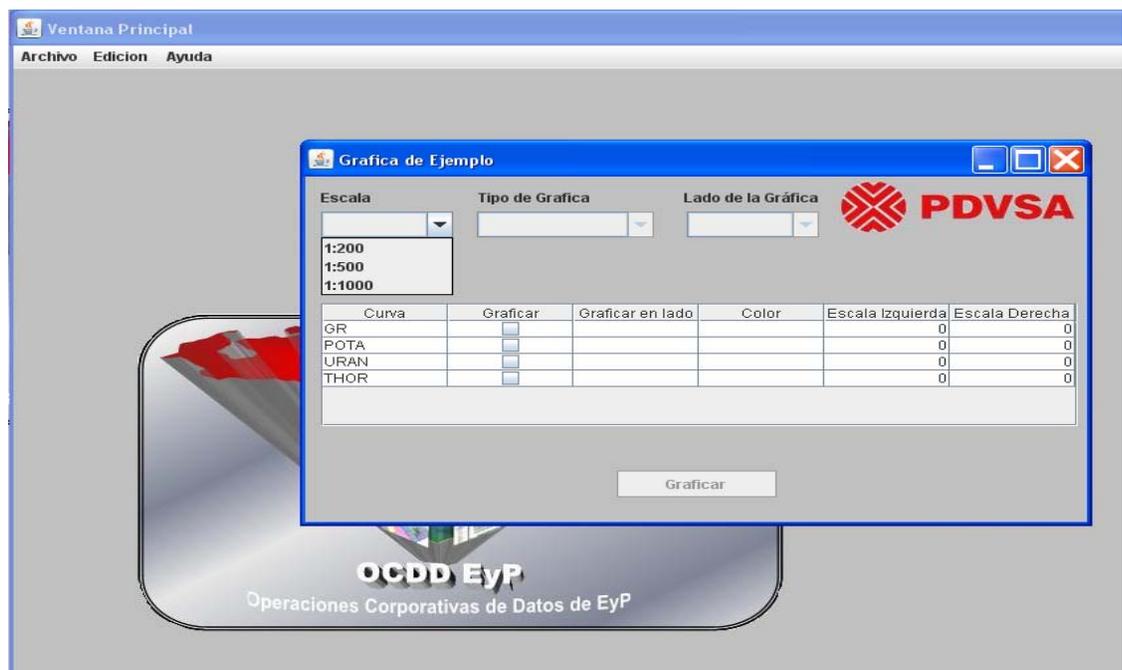


Figura 5.14 Ventana Graficar. Fuente Propia.

Nombre del Archivo: **Graficar.java**

```

/*
 * Graficar.java
 *
 * Created on 10/03/2009, 08:10:05 AM
 */
package graficador;

import java.awt.Color;
import java.io.IOException;
import java.util.ArrayList;
import javax.swing.JColorChooser;
import javax.swing.JOptionPane;
import javax.swing.table.TableColumn;
import javax.swing.table.TableColumnModel;
/**
 *
 * @author Julio Martinez
 */
public class Graficar extends javax.swing.JFrame {
    Object Cur[][] , Tit[]={"Curva","Graficar","Graficar en
lado","Color","Escala Izquierda","Escala Derecha"};
    int col=-1,fil=-1;
    Object Aux[][];
    java.io.File Archivo;
    boolean ban=false,verificar[];
    boolean act;
    boolean NoGraficar=false;
    int cont=0;
    /**Almacena los colores mostrados en la tabla, para su
redibujado*/
    Color colores[];
    private String TipoGrafica;

```

```

/**Alamacena la informaci3n de las curvas.*/
InformacionCurvas Datos;
/**Almacen la informaci3n del encabezado.*/
InformacionEncabezado encabezado;
/**Alamacena la informaci3n de los estudios.*/
private ArrayList<Estudio> Curvas;
/** Creates new form Graficar */
public Graficar()
{
    initComponents();
    //Reserva un espacio de memoria para la informaci3n de
las curvas
    System.gc(); System.gc();
    Datos=new InformacionCurvas();
    //Reserva un espacio de memoria para leer el encabezado
    encabezado=new InformacionEncabezado();
}
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
private void initComponents() {

    Abrir = new javax.swing.JFileChooser();
    ColorCur = new javax.swing.JColorChooser();
    jPanel1 = new javax.swing.JPanel();
    Escala = new javax.swing.JComboBox();
    jLabel4 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    Tipo = new javax.swing.JComboBox();
    jLabel2 = new javax.swing.JLabel();
    Lado = new javax.swing.JComboBox();
    jLabel3 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    Tabla = new javax.swing.JTable();
    Graficar = new javax.swing.JButton();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

        setBackground(java.awt.Color.lightGray);
        setBounds(new java.awt.Rectangle(400, 300, 0, 0));
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent
evt) {
                Cerrando(evt);
            }
        });

        jPanel1.setBackground(java.awt.Color.lightGray);

        Escala.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "1:200", "1:500", "1:1000" }));
        Escala.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                EscalaActionPerformed(evt);
            }
        });

        jLabel4.setText("Escala");

        jLabel1.setText("Tipo de Grafica");

        Tipo.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Lineal-Lineal", "Lineal-Logaritmica" }));
        Tipo.setEnabled(false);
        Tipo.addActionListener(new
java.awt.event.ActionListener() {

```

```

        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            TipoActionPerformed(evt);
        }
    });

    jLabel2.setText("Lado de la GrÃ¡fica");

    Lado.setEnabled(false);
    Lado.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            LadoActionPerformed(evt);
        }
    });

    jLabel3.setIcon(new javax.swing.ImageIcon("C:\\Documents
and Settings\\Julio C\\Escritorio\\Graficador\\Imagen\\PDVSA.jpg"));
    // NOI18N

    Tabla.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null}
        },
        new String [] {
            "Title 1"
        }
    ));
    Tabla.setEnabled(false);
    Tabla.setRowSelectionAllowed(false);
    Tabla.addMouseListener(new java.awt.event.MouseAdapter()
{
    public void mouseClicked(java.awt.event.MouseEvent
evt) {

```

```

        TablaMouseClicked(evt);
    }
});
jScrollPane.setViewportView(Tabla);

Graficar.setText("Graficar");
Graficar.setEnabled(false);
Graficar.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        GraficarActionPerformed(evt);
    }
});

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(236, 236, 236)
                .addComponent(Graficar,
javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(

```

```

.addGroup(jPanellLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
                .addComponent(jScrollPane)

.addGroup(jPanellLayout.createSequentialGroup())

.addGroup(jPanellLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
                .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE,          67,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(Escala,
javax.swing.GroupLayout.PREFERRED_SIZE,          100,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanellLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(jPanellLayout.createSequentialGroup())
                .addGap(19, 19, 19)
                .addComponent(jLabel1))

.addGroup(jPanellLayout.createSequentialGroup())
                .addGap(18, 18, 18)
                .addComponent(Tipo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(23, 23, 23)

.addGroup(jPanellLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING)
                .addComponent(jLabel2)

```

```

        .addComponent(Lado,
javax.swing.GroupLayout.PREFERRED_SIZE,          100,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel3))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())
        .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(jLabel1)
        .addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(Escala,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(Tipo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(Lado,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addComponent(jLabel3))
        .addGap(64, 64, 64)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(43, 43, 43)
        .addComponent(Graficar)
        .addContainerGap(124, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
)
        .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
)

```

```

        .addComponent(jPanell,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void TipoActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_TipoActionPerformed
        String opcion=(String)Tipo.getSelectedItem();
        if(opcion!=null)
        {
            if(opcion.contentEquals("Lineal-Lineal"))
            {
                Lado.removeAllItems();
                Lado.addItem("Derecho");
                Lado.addItem("Izquierdo");
            }
            else
            {
                Lado.removeAllItems();
                Lado.addItem("Lineal");
                Lado.addItem("Logaritmico");
            }
            Escala.setEnabled(false);
            Tipo.setEnabled(false);
            Lado.setSelectedIndex(-1);
            Tabla.setEnabled(true);
        }
    } //GEN-LAST:event_TipoActionPerformed
    /**En esta funci3n se busca el color*/
    private void TablaMouseClicked(java.awt.event.MouseEvent
evt) { //GEN-FIRST:event_TablaMouseClicked
        //        ban=true;

```

```

col=Tabla.getSelectedColumn();
fil=Tabla.getSelectedRow();
if(col==1)//booleano de la grafica
{
    verificar[fil]=(Boolean)Tabla.getValueAt(fil, col);
    System.out.println("el valor de la casilla es
"+verificar[fil]);
    act=false;
    if(verificar[fil]==false)
    {
        Tabla.setValueAt("", fil, 2);
        Tabla.setValueAt(0, fil, 4);
        Tabla.setValueAt(0, fil, 5);
        Lado.setEnabled(false);
        for(int j=0; j<verificar.length; j++)
        {
            if(!verificar[j])
            {
                cont++;
                //          act=false;
                //          break;
            }
        }
        if(cont==verificar.length)
        {
            act = true;
        }
        cont=0;
        //  Escala.setEnabled(true);
        if(act)
        {
            Escala.setEnabled(true);
            Escala.setSelectedIndex(-1);
            Tipo.setSelectedIndex(-1);
            Tipo.setEnabled(false);
        }
    }
}

```

```

        Lado.setSelectedIndex(-1);
        Lado.setEnabled(false);
        Tabla.setEnabled(false);
        Graficar.setEnabled(false);
    }
}
else
{
    if(col==1)
    {
        if(Tabla.getValueAt(fil,
0).toString().toUpperCase().contentEquals("TENS"))
        {
            Tabla.setValueAt((Object)"Centro", fil,
2);
        }
    }
}
}
if(col==2)// columna correspondiente al lado de la
grafica
{
    if(!Tabla.getValueAt(fil,
0).toString().toUpperCase().contentEquals("TENS"))
    {
        if(verificar[fil]==true)
            Lado.setEnabled(true);
        else
            Lado.setEnabled(false);
    }
}
if(col==3)// columna de seleccion de color
{
    if(verificar[fil]==true)
    {

```

```

        Lado.setEnabled(false);
        Color background =
JColorChooser.showDialog(null, "Cambiar Color de la
Curva", Color.WHITE);
        if(background!=Color.WHITE)
        {
            TableColumnModel modeloColumna=
Tabla.getColumnModel();
            TableColumn columna =
modeloColumna.getColumn(Tabla.getSelectedColumn());
            colores[fil]=background;
            columna.setCellRenderer(new
Render(Tabla.getSelectedRow(), colores));
            Tabla.setColumnModel(modeloColumna);
            Tabla.repaint();
        }
        else
        {

javax.swing.JOptionPane.showMessageDialog(null, "No se selecciono
ningun color", "Seleccion de Archivos", 3);

        }
        if(Tabla.getValueAt(fil, 2).toString()!=null)
            if(Tabla.getValueAt(fil,
2).toString().contentEquals("Logaritmico"))
            {
                Tabla.setValueAt(0.2, fil, 4);
                Tabla.setValueAt(2000, fil, 5);
                Graficar.setEnabled(true);
            }
        }

    }
    if(col==4 | col==5)

```

```

        { // activa el boton de graficar
            Graficar.setEnabled(true);

        }

        TipoGrafica=(String) Tipo.getSelectedItem();//Se guarda
el tipo de la grafica
    } //GEN-LAST:event_TablaMouseClicked

    private void LadoActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_LadoActionPerformed
        if(col!=-1&&fil!=-1&&verificar[fil]==true)
        {
            if(col==2)
            {
                Tabla.setValueAt(Lado.getSelectedItem(),
fil, col);

                Lado.setSelectedIndex(-1);
                Lado.setEnabled(false);
            }
            else

            javax.swing.JOptionPane.showMessageDialog(null,"Selecciono la celda
            equivocada","Seleccion de Celdas",3);
        }
    } //GEN-LAST:event_LadoActionPerformed

    private void GraficarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
    FIRST:event_GraficarActionPerformed
        try
        {
            encabezado.EstudioUnidades(verificar.length);
        }
        catch (IOException Ex)
        {

```

```

        JOptionPane.showMessageDialog(null,"Error: con el
sistema de coordenadas "+Ex.getMessage(),
"AtenciÃ³n",JOptionPane.ERROR_MESSAGE);
    }
    //Obtiene las unidades de los estudios en el encabezado
    String Und[][]=encabezado.ObtenerUnidades();
    for(int i=0; i<Curvas.size();i++)
    {
        Estudio nodo=new Estudio();
        nodo=Curvas.get(i);
        for(int j=0; j<Und.length && nodo!=null; j++)
        {
            if(nodo.ObtenerNombre().equals(Und[j][0]))
            {
                nodo.AsignarUnidad(Und[j][1]);
                Curvas.set(i, nodo);
            }
        }
    }

//*****
***

    int n=Curvas.size(), j=0;
    for(int i=0; i<n;i++)
    {
        Estudio nodo=new Estudio();
        nodo=Curvas.get(i);
        if((Boolean)Tabla.getValueAt(i, 1)==true)
        {
            nodo.AsignarColor(colores[i]); //color de la
grafica
            nodo.AsignarLado((String)Tabla.getValueAt(i,
2)); //Lado de la grafica
            nodo.AsignarRangos(Double.valueOf(Tabla.getValueAt(i,

```

```

4).toString()), Double.valueOf(Tabla.getValueAt(i, 5).toString()));
//rangos izq y der
        Curvas.set(i, nodo);
    }
}
for(int i=n-1; i>=0;i--)
{
    if((Boolean)Tabla.getValueAt(i, 1)!=true)
    {
        Curvas.remove(i);
    }
}
// Enviar escala
int Esc=0;
if(Escala.getSelectedIndex()==0)
{
    Esc=2;
}
else if(Escala.getSelectedIndex()==1)
{
    Esc=10;
}
else
{
    Esc=20;
}
System.gc(); System.gc();
//LLama a la ventana de la grafica
if(TipoGrafica.contentEquals("Lineal-Lineal"))
{
    VentanaGrafica vent=new VentanaGrafica(Curvas,
encabezado.ObtenerNulo(), encabezado.ObtenerDivision(), Esc);
    vent.setBounds(270, 458, 960, 550);
    vent.setVisible(true);
}
}

```

```

else
{
    VentanaGraficaLog vent = new VentanaGraficaLog(Curvas,
encabezado.ObtenerNulo(), encabezado.ObtenerDivision(), Esc);
    vent.setTitle("Grafica "+Archivo.getName());
    vent.setBounds(270, 458, 960, 550);
    vent.setVisible(true);
}
this.dispose();

} //GEN-LAST:event_GraficarActionPerformed

private void
EscalaActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_EscalaActionPerformed
    Tipo.setEnabled(true);
} //GEN-LAST:event_EscalaActionPerformed
private void Cerrando(java.awt.event.WindowEvent evt)
{ //GEN-FIRST:event_Cerrando
    //Liberando memoria de la maquina virtual
    System.gc();
    System.gc();
    System.gc();
    System.gc();
} //GEN-LAST:event_Cerrando
/**Esta funci3n se encarga de crear el modelo de la tabla y
mostrarlo en la
* ventana.
* @param NombreArchivo Tipo File, recibe el archivo ha
utilizar.
*/
public void mostrarcuvas(java.io.File NombreArchivo)
{
    Archivo=NombreArchivo;
    String nomarc;

```

```

nomarc=NombreArchivo.getName();
nomarc=nomarc.replace(".las","");
this.setTitle("Grafica de "+nomarc);
System.gc(); System.gc();
//Lee el encabezado del archivo
encabezado.Recibir_Archivo(Archivo);
//Selecciona las curvas del paquete
Datos.AsignarNulo(encabezado.ObtenerNulo());
if(Datos.Recibir_Archivo(Archivo))
{
    System.gc(); System.gc();
    //Escoge el archivo del cual se va ha graficar
    Curvas=Datos.ObtenerEstudios(); //Obtiene toda la
informaci3n de las curvas
    //Arreglo de control de las curvas seleccionadas
    verificar = new boolean [Curvas.size()];
    //Reservar el espacio en memoria para los render del
color*****
    colores= new Color [Curvas.size()];
    for(int i=0; i<Curvas.size(); i++)
    {
        colores[i]=Color.white;
    }
    Cur = new Object[Curvas.size()][6];
    for(int i=0;i<Curvas.size();i++)
    {
        Cur[i][0] =Curvas.get(i).ObtenerNombre();
        Cur[i][4] = 0;
        Cur[i][5] = 0;
    }
    Tabla.setModel(new
javax.swing.table.DefaultTableModel(Cur,Tit)
{
    Class[] types = new Class []
    {

```

```

        java.lang.Object.class,
java.lang.Boolean.class,                java.lang.Object.class,
java.lang.Object.class,                java.lang.Double.class,
java.lang.Double.class
    };
    @Override
    public Class getColumnClass(int columnIndex)
    {
        return types [columnIndex];
    }
    boolean[] canEdit = new boolean []
    {
        false, true, false, false, true, true
    };
    @Override
    public boolean isCellEditable(int rowIndex, int
columnIndex)
    {
        return canEdit [columnIndex];
    }
    });
    for(int i=0; i<Curvas.size(); i++)
    {
        Tabla.setValueAt(false, i, 1);
    }
    Escala.setSelectedIndex(-1);
    Tipo.setSelectedIndex(-1);
    Tipo.setEnabled(false);
    }
    else
    {
        NoGraficar=true;
        dispose();
    }
}

```

```
public boolean SioNo()
{
    return NoGraficar;
}
// Variables declaration - do not modify//GEN-
BEGIN:variables
private javax.swing.JFileChooser Abrir;

private javax.swing.JColorChooser ColorCur;

private javax.swing.JComboBox Escala;

private javax.swing.JButton Graficar;

private javax.swing.JComboBox Lado;

private javax.swing.JTable Tabla;

private javax.swing.JComboBox Tipo;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

// End of variables declaration//GEN-END:variables
}
```

5.2.4 Implementación del caso de uso Editar Archivo de datos

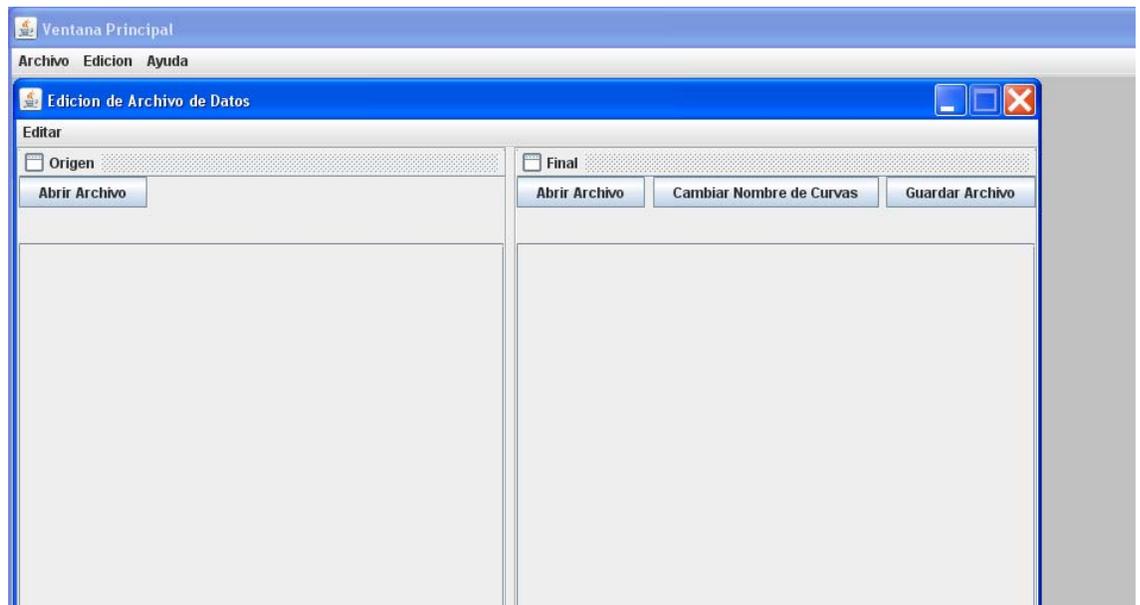


Figura 5.15 Ventana de Edición de archivo de datos. Fuente Propia.

Nombre del Archivo: `Editar.java`

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * Editar.java
 *
 * Created on 13/03/2009, 10:07:08 AM
 */

package graficador;

```

```

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;

/**
 *
 * @author Julio Martinez
 */

public class Editor extends javax.swing.JFrame {
    java.io.File NombreArchivo,NombreArchivo2,NombreArchivo3;
    String nomarc;
    Object Edit[][];
    double nulo;
    boolean copiar=false;
    int col[],fil[],maxfil,maxcol;
    Object Cur[][][],Tit[],CurAux[][][],Curmod[][];
    java.io.BufferedReader streami;
    InformacionCurvas Datos;
    /**Almacen la informaci3n del encabezado.**/
    InformacionEncabezado encabezado;
    /** Creates new form Editor */
    public void Editor() {
        initComponents();
        Datos = new InformacionCurvas();
        Filtro filtro = new Filtro();
        /* extensiones que quieres aadir*/
        filtro.addExtension("las");
        filtro.addExtension("LAS");
        /* descripcion del filtro*/
        filtro.setDescription("Archivos LAS");
        Abrir.addChoosableFileFilter(filtro);
        Guardar.addChoosableFileFilter(filtro);
        System.gc(); System.gc();
    }
}

```

```

    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this
method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        Abrir = new javax.swing.JFileChooser();
        Guardar = new javax.swing.JFileChooser();
        CambioNombreTitulo = new javax.swing.JFrame();
        jScrollPane3 = new javax.swing.JScrollPane();
        jTable3 = new javax.swing.JTable();
        GuardarTitulo = new javax.swing.JButton();
        VentanaIzquierda = new javax.swing.JInternalFrame();
        Tabla1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        Abrir1 = new javax.swing.JButton();
        NombreArch1 = new javax.swing.JLabel();
        VentanaDerecha = new javax.swing.JInternalFrame();
        Abrir2 = new javax.swing.JButton();
        Guardar2 = new javax.swing.JButton();
        Tabla2 = new javax.swing.JScrollPane();
        jTable2 = new javax.swing.JTable();
        NombreArch2 = new javax.swing.JLabel();
        CambiarNombreTit = new javax.swing.JButton();
        MenuBar = new javax.swing.JMenuBar();
        Editar = new javax.swing.JMenu();
        Copiar = new javax.swing.JMenuItem();
        Pegar = new javax.swing.JMenuItem();

```

```

Abrir.setAcceptAllFileFilterUsed(false);
Filtro filtro=new Filtro();
filtro.addExtension("las");
filtro.setDescription("Archivos LAS");

Guardar.setAcceptAllFileFilterUsed(false);
Guardar.setApproveButtonText("Guardar");
Guardar.setEnabled(false);
Guardar.setName(""); // NOI18N

CambioNombreTitulo.setDefaultCloseOperation(javax.swing.WindowConsta
nts.DISPOSE_ON_CLOSE);

CambioNombreTitulo.setTitle("Cambio de Nombre de
Curvas");

jScrollPane3.setAutoscrolls(true);

jTable1.setVisible(false);
jTable3.setModel(new
javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null}
    },
    new String [] {
        ""
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];

```

```

        }
    });

jTable3.setAutoResizeMode( javax.swing.JTable.AUTO_RESIZE_OFF);
    jTable3.setColumnSelectionAllowed(true);
    jScrollPane3.setViewportViewView(jTable3);

    GuardarTitulo.setText("Guardar");
    GuardarTitulo.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            GuardarTituloActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout CambioNombreTituloLayout = new
javax.swing.GroupLayout(CambioNombreTitulo.getContentPane());

CambioNombreTitulo.getContentPane().setLayout(CambioNombreTituloLayo
ut);

    CambioNombreTituloLayout.setHorizontalGroup(

CambioNombreTituloLayout.createParallelGroup( javax.swing.GroupLayout
.Alignment.LEADING)

.addGroup(CambioNombreTituloLayout.createSequentialGroup()
        .addComponent(GuardarTitulo)
        .addContainerGap()
        .addComponent(jScrollPane3,
javax.swing.GroupLayout.DEFAULT_SIZE, 590, Short.MAX_VALUE)
    );
    CambioNombreTituloLayout.setVerticalGroup(

```

```

CambioNombreTituloLayout.createParallelGroup( javax.swing.GroupLayout
.Alignment.LEADING)

.addGroup(CambioNombreTituloLayout.createSequentialGroup()
        .addComponent( jScrollPane3,
javax.swing.GroupLayout.PREFERRED_SIZE,          54,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent( GuardarTitulo)

.addContainerGap( javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        );

setDefaultCloseOperation( javax.swing.WindowConstants.DISPOSE_ON_CLOS
E);

        setTitle("Edicion de Archivo de Datos");
        setResizable(false);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent
evt) {
                Cerrando(evt);
            }
        });
}

VentanaIzquierda.setBorder( javax.swing.BorderFactory.createEtchedBor
der());

VentanaIzquierda.setTitle("Origen");
VentanaIzquierda.setAutoScrolls(true);
try {
    VentanaIzquierda.setSelected(true);
}

```

```

    } catch (java.beans.PropertyVetoException e1) {
        e1.printStackTrace();
    }
    VentanaIzquierda.setVisible(true);

    Tabla1.setAutoscrolls(true);

    jTable1.setVisible(false);
    jTable1.setModel(new
javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null}
        },
        new String [] {
            ""
        }
    ));

    jTable1.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
    jTable1.setColumnSelectionAllowed(true);
    Tabla1.setViewportView(jTable1);

    Abrir1.setText("Abrir Archivo");
    Abrir1.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            Abrir1ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout VentanaIzquierdaLayout = new
javax.swing.GroupLayout(VentanaIzquierda.getContentPane());

VentanaIzquierda.getContentPane().setLayout(VentanaIzquierdaLayout);

```

```

VentanaIzquierdaLayout.setHorizontalGroup(

VentanaIzquierdaLayout.createParallelGroup( javax.swing.GroupLayout.A
lignment.LEADING)

.addGroup(VentanaIzquierdaLayout.createSequentialGroup()
        .addComponent(Abrir1)
        .addContainerGap(301, Short.MAX_VALUE))
        .addComponent(NombreArch1,
javax.swing.GroupLayout.DEFAULT_SIZE, 408, Short.MAX_VALUE)
        .addComponent(Tabl1,
javax.swing.GroupLayout.DEFAULT_SIZE, 408, Short.MAX_VALUE)
        );
VentanaIzquierdaLayout.setVerticalGroup(

VentanaIzquierdaLayout.createParallelGroup( javax.swing.GroupLayout.A
lignment.LEADING)

.addGroup(VentanaIzquierdaLayout.createSequentialGroup()
        .addComponent(Abrir1)

.addPreferredGap( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(NombreArch1,
javax.swing.GroupLayout.PREFERRED_SIZE,                                18,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(Tabl1,
javax.swing.GroupLayout.DEFAULT_SIZE, 816, Short.MAX_VALUE))
        );

VentanaDerecha.setBorder( javax.swing.BorderFactory.createEtchedBorde
r());

VentanaDerecha.setTitle("Final");

```

```

VentanaDerecha.setAutoscrolls(true);
try {
    VentanaDerecha.setSelected(true);
} catch (java.beans.PropertyVetoException e1) {
    e1.printStackTrace();
}
VentanaDerecha.setVisible(true);

Abrir2.setText("Abrir Archivo");
Abrir2.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        Abrir2ActionPerformed(evt);
    }
});

Guardar2.setText("Guardar Archivo");
Guardar2.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        Guardar2ActionPerformed(evt);
    }
});

jTable2.setVisible(false);
jTable2.setModel(new
javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null}
    },
    new String [] {
        ""
    }

```

```

    ));

jTable2.setAutoResizeMode( javax.swing.JTable.AUTO_RESIZE_OFF);
    jTable2.setCellSelectionEnabled(true);
    Tabla2.setViewportView(jTable2);

    CambiarNombreTit.setText("Cambiar Nombre de Curvas");
    CambiarNombreTit.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed( java.awt.event.ActionEvent evt) {
            CambiarNombreTitActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout VentanaDerechaLayout = new
javax.swing.GroupLayout(VentanaDerecha.getContentPane());

VentanaDerecha.getContentPane().setLayout(VentanaDerechaLayout);
    VentanaDerechaLayout.setHorizontalGroup(

VentanaDerechaLayout.createParallelGroup( javax.swing.GroupLayout.Ali
gnment.LEADING)

.addGroup(VentanaDerechaLayout.createSequentialGroup()
        .addComponent(Abrir2)

.addPreferredGap( javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(CambiarNombreTit)

.addPreferredGap( javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(Guardar2)

```

```

        .addComponent(NombreArch2,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 437, Short.MAX_VALUE)
        .addComponent(Tabla2,
javax.swing.GroupLayout.DEFAULT_SIZE, 437, Short.MAX_VALUE)
    );
    VentanaDerechaLayout.setVerticalGroup(

VentanaDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

.addGroup(VentanaDerechaLayout.createSequentialGroup())

.addGroup(VentanaDerechaLayout.createParallelGroup(javax.swing.Group
Layout.Alignment.BASELINE)
        .addComponent(Abrir2)
        .addComponent(Guardar2)
        .addComponent(CambiarNombreTit))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(NombreArch2,
javax.swing.GroupLayout.PREFERRED_SIZE, 18,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(Tabla2,
javax.swing.GroupLayout.DEFAULT_SIZE, 816, Short.MAX_VALUE))
    );

    Editar.setText("Editar");

Copiar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.ev
ent.KeyEvent.VK_C, java.awt.event.InputEvent.CTRL_MASK));
    Copiar.setText("Copiar");

```

```

        Copiar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                CopiarActionPerformed(evt);
            }
        });
        Editor.add(Copiar);

Pegar.setAccelerator( javax.swing.KeyStroke.getKeyStroke( java.awt.eve
nt.KeyEvent.VK_V, java.awt.event.InputEvent.CTRL_MASK));
        Pegar.setText( "Pegar" );
        Pegar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                PegarActionPerformed(evt);
            }
        });
        Editor.add(Pegar);

        MenuBar.add(Editor);

        setJMenuBar( MenuBar );

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout( getContentPane() );
        getContentPane().setLayout( layout );
        layout.setHorizontalGroup(

layout.createParallelGroup( javax.swing.GroupLayout.Alignment.LEADING
)

        .addGroup( layout.createSequentialGroup( )

```

```

        .addComponent(VentanaIzquierda,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(VentanaDerecha))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent(VentanaIzquierda,
javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(VentanaDerecha,
javax.swing.GroupLayout.Alignment.TRAILING)
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

@SuppressWarnings("static-access")
private void
AbrirActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_AbrirActionPerformed
    // TODO add your handling code here:
    try
    {
        int returnVal = Abrir.showOpenDialog(this);
        if (returnVal == Abrir.APPROVE_OPTION)
        {
            NombreArchivo = Abrir.getSelectedFile();
            nomarc = NombreArchivo.getName();
            nomarc = nomarc.toUpperCase();
            nomarc = nomarc.replace(".LAS", "");

```

```

//*****
        System.gc(); System.gc();
        encabezado=new InformacionEncabezado();
        encabezado.Recibir_Archivo(NombreArchivo);
        Datos.AsignarNulo(encabezado.ObtenerNulo());
        if(Datos.Recibir_Archivo(NombreArchivo))
        {
            System.gc(); System.gc();
            NombreArch1.setText(nomarc);
            NombreArch1.setVisible(true);
            jTable1.setModel(new
javax.swing.table.DefaultTableModel(Datos.ObtenerCoordenadas(),
Datos.ObtenerTitulos()));
            jTable1.setVisible(true);
        }
    }
}
catch (Exception e)
{
    javax.swing.JOptionPane.showMessageDialog(null,"No
se selecciono ningun archivo"+e.getMessage(),"Seleccion de
Archivos",3);
}
} //GEN-LAST:event_Abrir1ActionPerformed

        @SuppressWarnings("static-access")
        private void
Abrir2ActionPerformed(java.awt.event.ActionEvent evt) //GEN-
FIRST:event_Abrir2ActionPerformed
        // TODO add your handling code here:

        try
        {

```

```

int returnVal = Abrir.showOpenDialog(this);
if (returnVal == Abrir.APPROVE_OPTION)
{
    NombreArchivo2 = Abrir.getSelectedFile();
    nomarc = NombreArchivo2.getName();
    nomarc = nomarc.toUpperCase();
    nomarc = nomarc.replace(".LAS", "");

//*****
    System.gc(); System.gc();
    encabezado=new InformacionEncabezado();
    encabezado.Recibir_Archivo(NombreArchivo);
    nulo=encabezado.ObtenerNulo();
    Datos.AsignarNulo(nulo);
    if(Datos.Recibir_Archivo(NombreArchivo))
    {
        System.gc(); System.gc();
        NombreArch2.setText(nomarc);
        NombreArch2.setVisible(true);
        Tit=Datos.ObtenerTitulos();
        jTable2.setModel(new
javax.swing.table.DefaultTableModel(Datos.ObtenerCoordenadas(),
Datos.ObtenerTitulos()));
        jTable2.setVisible(true);
    }
}
catch (Exception e)
{

javax.swing.JOptionPane.showMessageDialog(null,"No se selecciono
ningun archivo"+e.getMessage(),"Seleccion de Archivos",3);
}

} //GEN-LAST:event_Abrir2ActionPerformed

```

```

        @SuppressWarnings("static-access")
        private void
Guardar2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_Guardar2ActionPerformed
        // TODO add your handling code here:
        boolean ban=true;
        int filas,columnas;
        String dato,curva;
        String auxiliar;
        int returnVal = Guardar.showSaveDialog(this);
        if (returnVal == Guardar.APPROVE_OPTION)
        {
            NombreArchivo3=Guardar.getSelectedFile();
            String nombreArchivo = NombreArchivo3.getPath();
            nombreArchivo.toLowerCase();
            if(!nombreArchivo.contains(".las"))
            {
                nombreArchivo=nombreArchivo.concat(".las");
            }
            try
            {
                File file = new File (nombreArchivo);
                FileWriter out = new FileWriter(file);
                streami=new java.io.BufferedReader(new
java.io.FileReader(NombreArchivo2.toString()));
                if(streami!=null)
                {
                    while((auxiliar=streami.readLine())!=null)
                    {
                        if(auxiliar.contains("~A"))
                        {
                            ban=false;
                        }
                    }
                    if(ban)

```

```

        {
            String text = auxiliar;
            out.write(text+'\n');
        }
    }
}
filas=jTable2.getRowCount();
columnas=jTable2.getColumnCount();
out.write("~A"+"      ");
for(int i=0;i<columnas;i++)
{
    curva=jTable2.getColumnName(i);
    out.write(curva+"      ");
}
out.write('\n');
for(int i=0;i<filas;i++)
{
    out.write('\t');
    for(int j=0;j<columnas;j++)
    {
dato=String.valueOf(jTable2.getValueAt(i, j));
        if(dato.contentEquals("null"))
        {
            out.write(nulo+"      ");
        }
        else
        {
            out.write(dato+"      ");
        }
    }
    out.write('\n');
}
out.close();
}

```

```

        catch (IOException e)
        {
            JOptionPane.showMessageDialog(null, "NO SE PUDO
GUARDAR " + nomarc+ " DEBIDO A " + e, "Error de
Escritura",JOptionPane.ERROR_MESSAGE);
        }
    }
} //GEN-LAST:event_Guardar2ActionPerformed

private void
CopiarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_CopiarActionPerformed
    col = jTable1.getSelectedColumns();
    fil = jTable1.getSelectedRows();
    System.gc(); System.gc();
    if(col.length==0)
    {
        javax.swing.JOptionPane.showMessageDialog(null, "NO
SELECCIONO CELDAS PARA COPIAR", "Error",JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        copiar=true;
        int x=0,y=0;
        Edit = new Object[fil.length][col.length];
        for(int i=fil[0];i<fil[0]+fil.length;i++)
        {
            for(int j=col[0];j<col[0]+col.length;j++)
            {
                Edit[y][x]= jTable1.getValueAt(i, j);
                x++;
            }
            x=0;
            y++;
        }
    }
}

```

```

    }
} //GEN-LAST:event_CopiarActionPerformed
private void PegarActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_PegarActionPerformed
    System.gc(); System.gc();
    if(copiar)
    {
        int x=0,y=0;
        int posx, posy, posxaux, posyaux;
        posx=jTable2.getSelectedColumn();
        posy=jTable2.getSelectedRow();
        posyaux=posy;
        maxfil=jTable2.getRowCount();
        maxcol=jTable2.getColumnCount();
        if(posx+Edit[0].length<=maxcol)
        {
            posxaux=posx;
            if(posx==-1 || posy==-1)
            {
                javax.swing.JOptionPane.showMessageDialog(null, "NO SELECCIONO LA
POSICION EN LA TABLA DESTINO DONDE SE VAN A COLOCAR LOS
DATOS", "Error", 3);
            }
            else
            {
                for(int i=fil[0];i<fil[0]+fil.length;i++)
                {
                    for(int
j=col[0];j<col[0]+col.length;j++)
                    {
                        if(posy==jTable2.getRowCount())
                        {
                            CurAux=
new
Object[(maxfil+Edit.length)-(maxfil-posyaux)][maxcol];

```

```

                                for(int
a=0;a<jTable2.getRowCount();a++)
                                {
                                    for(int
b=0;b<jTable2.getColumnCount();b++)
CurAux[a][b]=jTable2.getValueAt(a, b);
                                    }
                                }
                                jTable2.setModel(new
javax.swing.table.DefaultTableModel(CurAux,Tit));
                                jTable2.setVisible(true);
                                }
                                if(x<jTable2.getColumnCount())
                                {
                                    jTable2.setValueAt(Edit[y][x],
posy, posxaux );

                                    x++;
                                    posxaux++;
                                }
                                }
                                posxaux=posx;
                                x=0;
                                y++;
                                posy++;
                                }
                                }
                                else
                                {
                                    JOptionPane.showMessageDialog(null, "NO SE PUEDE
PEGAR LA SELECCION EN LA TABLA DESTINO", "Error de
Escritura",JOptionPane.ERROR_MESSAGE);
                                }
                                }
                                else

```

```

        {
            JOptionPane.showMessageDialog(null, "NO COPIO
CELDAS", "Error",JOptionPane.ERROR_MESSAGE);
        }
    }//GEN-LAST:event_PegarActionPerformed
    private void
CambiarNombreTitActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_CambiarNombreTitActionPerformed
    System.gc(); System.gc();
    Curmod=new Object[1][Tit.length];
    Curmod[0]=Tit;
    jTable3.setModel(new
javax.swing.table.DefaultTableModel(Curmod,Tit));
    CambioNombreTitulo.setBounds(100, 100, Tit.length*75,
150);
    CambioNombreTitulo.setVisible(true);
} //GEN-LAST:event_CambiarNombreTitActionPerformed
    private void
GuardarTituloActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_GuardarTituloActionPerformed
    System.gc(); System.gc();
    for(int i=1;i<Tit.length;i++)
    { Tit[i]=jTable3.getValueAt(0, i); }
    jTable2.setModel(new
javax.swing.table.DefaultTableModel(Datos.ObtenerCoordenadas(),Tit))
;
    jTable2.setVisible(true);
    CambioNombreTitulo.dispose();
} //GEN-LAST:event_GuardarTituloActionPerformed
    private void Cerrando(java.awt.event.WindowEvent evt) { //GEN-
FIRST:event_Cerrando
        //Liberando memoria de la maquina virtual
        System.gc(); System.gc();
        System.gc();
        System.gc();

```

```

} //GEN-LAST:event_Cerrando
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Editor().setVisible(true);
        }
    });
}
// Variables declaration - do not modify //GEN-
BEGIN:variables
    private javax.swing.JFileChooser Abrir;
    private javax.swing.JButton Abrir1;
    private javax.swing.JButton Abrir2;
    private javax.swing.JButton CambiarNombreTit;
    private javax.swing.JFrame CambioNombreTitulo;
    private javax.swing.JMenuItem Copiar;
    private javax.swing.JMenu Editor;
    private javax.swing.JFileChooser Guardar;
    private javax.swing.JButton Guardar2;
    private javax.swing.JButton GuardarTitulo;
    private javax.swing.JMenuBar MenuBar;
    private javax.swing.JLabel NombreArch1;
    private javax.swing.JLabel NombreArch2;
    private javax.swing.JMenuItem Pegar;
    private javax.swing.JScrollPane Tabla1;
    private javax.swing.JScrollPane Tabla2;
    private javax.swing.JInternalFrame VentanaDerecha;
    private javax.swing.JInternalFrame VentanaIzquierda;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JTable jTable1;
    private javax.swing.JTable jTable2;
    private javax.swing.JTable jTable3;

```

```
// End of variables declaration//GEN-END:variables
}
```

5.3 Pruebas

El objetivo principal de llevar a cabo pruebas, es asegurarse que todas las ramificaciones del sistema, hasta las ventanas mas específicas de la aplicación funcionen y no presenten ningún problema.

Se catalogaron los posibles errores del sistema con el propósito de obtener la versión beta del Sistema JGrafep. Para efectos de la explicación sólo serán mostrados algunos de los errores que pudieran surgir en la manipulación del software JGrafep.

Tabla 5.1 Descripción de los mensajes de error (1/2). Fuente propia.

| Mensaje de Error | Descripción |
|--|--|
| NO SE PUEDE PEGAR LA SELECCION EN LA TABLA DESTINO. | El usuario esta en el proceso de edición de archivo de datos, y la selección hecha en el archivo origen no es compatible con el archivo destino. |
| NO COPIO CELDAS | El usuario esta en el proceso de edición de archivo de datos, y pulso el botón Pegar, sin haber hecho una selección en el archivo origen y haber pulsado el botón de Copiar. |
| NO SELECCIONO LA POSICION EN LA TABLA DESTINO DONDE SE VAN A COLOCAR LOS DATOS | El usuario esta en el proceso de edición de archivo de datos, y después de haber copiado las celdas en el archivo origen, pulso el botón pegar, sin haber seleccionado la celda destino. |
| NO SELECCIONO CELDAS PARA COPIAR | El usuario esta en el proceso de edición de archivo de datos, y pulso el botón Pegar, sin haber hecho una selección en el archivo origen ni haber pulsado el botón de Copiar. |

Tabla 5.1 Descripción de los mensajes de error (2/2). Fuente propia.

| Mensaje de Error | Descripción |
|------------------------------|--|
| NO SE PUDO GUARDAR | El usuario esta en el proceso de edición de archivo de datos e intento Guardar el nuevo archivo sin que la acción fuese exitosa. |
| ERROR: AL GRAFICAR LOS EJES. | El usuario esta en el proceso de Graficación, y no se pueden crear los ejes de graficación por error en el archivo de datos. |

CONCLUSIONES

1. Se efectuó con éxito un estudio detallado de todo el procedimiento y los requerimientos asociados a la verificación de la información de los registros de pozos en el Departamento de Exploración y Producción de la empresa PDVSA, realizándose eficazmente el almacenamiento y graficación de la información recabada en la Gerencia de Operaciones de Datos.
2. Se establecieron óptimamente los datos de entrada necesarios para la elaboración del Software, satisfaciendo así, las necesidades de los usuarios en la gerencia de Operaciones de Datos, en sus distintos departamentos: memoria corporativa y manejo del dato.
3. Se realizó la descripción y especificación de los requisitos del sistema, basándose en las necesidades y requerimientos de la GODD para el óptimo desempeño de su personal al utilizar la aplicación.
4. Se realizó de manera eficiente y clara el diseño de los módulos que conforman el sistema, logrando una arquitectura estable por la cual es sustentado.
5. Fueron realizadas pruebas de manera individual a cada uno de los procesos, mitigando de esta forma los fallos que estos pudiesen presentar, evitando así, inconvenientes con los usuarios que están familiarizándose con el sistema.

6. La Metodología Orientada a Objetos basándose en el ciclo de vida del Software proporcionó un marco de trabajo ideal para el diseño y la construcción por etapas del sistema.

RECOMENDACIONES

1. Realizar un mantenimiento periódico a los equipos que tengan instalado el sistema.
2. Desarrollar un módulo que permita, a partir de los archivos de datos LIS (Log Information Standard) y DLIS (Digital Log Interchange Standard), poder realizar la creación de archivos LAS.
3. Realizar cursos de adiestramiento para los usuarios que van a interactuar con el sistema.
4. Se le sugiere al usuario del sistema leer el manual de usuario antes de manipular el software.

BIBLIOGRAFÍA

1. Presilla, R. (2009). “Software de apoyo para la determinación y jerarquización de unidades de flujo, en función de índices de calidad de roca y saturación de fluidos producibles en yacimientos hidrocarburíferos, en la empresa INTER-ROCK. C.A.”, Trabajo de grado presentado en la Universidad de Oriente, Núcleo Anzoátegui.
2. Cardozo, D. (2007). “Desarrollo de un sistema que permita automatizar la información del historial de pozo, en la unidad de explotación de yacimientos de PDVSA-SANTOME”. Trabajo de grado presentado en la Universidad de Oriente, Núcleo Anzoátegui.
3. Coa, L. (2007). “Desarrollo de un sistema que permita automatizar la información del historial de pozo, en la unidad de explotación del yacimiento de PDVSA – San Tome”. Trabajo de grado presentado en la Universidad de Oriente, Núcleo Anzoátegui.
4. Prof. Nestor V. Queipo, Ph.D. (2006). “Taller de Exploración y Producción desde una perspectiva centrada en los datos”. Universidad del Zulia. Facultad de Ingeniería. Instituto de Calculo Aplicado.
5. Ferré G., Xavier (Univ. Politécnica de Madrid - España) y Sánchez S., María I. (Univ. Carlos III de Madrid - España) (2004). **Desarrollo Orientado a Objetos con UML.**

6. London, M. (2004). “Desarrollo de un Software que permita el Monitoreo de la Información de Estado de un Portal Web Alojado en la Intranet Corporativa de PDVSA, utilizando la Nueva Plataforma de Microsoft.NET”. Trabajo de grado presentado en la Universidad de Oriente, Núcleo Anzoátegui.
7. Barreto, H. (2003). “Diseño De Un Sistema De Información De Inventario En El Departamento IT (Información Tecnología) de una Empresa Petrolera. Puerto La Cruz, Sector Venecia”. Trabajo de grado presentado en la Universidad de Oriente Núcleo de Anzoátegui.
8. Stallins, William. (2000). Sistemas Operativos, Prentice Hall. México. 2da Edición
9. Deitel, H.M. Deitel, P.J. (1998). Cómo programar en Java, Prentice Hall. México.
10. Letelier T., Patricio. (1998). Desarrollo de Software Orientado a Objeto usando UML. Universidad Politécnica de Valencia - España. Departamento Sistemas Informáticos y Computación.
11. Linda Gail Christie/ Jhon Christie (1984). Enciclopedia de términos de computación Prentice Hall Hispanoamericana S.A. Mexico.

APENDICE A MANUAL DE USUARIO

A.1 Instalación del programa.

Para instalar el programa se debe:

Se debe Colocar el CD de instalación en la unidad de CD-ROM y ejecutar el paquete de Instalación, haciendo doble clic en el icono que se encuentra dentro del CD con el nombre “Instalacion.exe”.

Una vez finalizada la ejecución del instalador, este crea una carpeta en la ubicación C:\Instalacion, luego se procede a instalar el programa JGrafep, este programa se encuentran dentro del directorio creado por el instalador. El programa será instalado en la carpeta C:\Archivos de programa\JGrafep.

Luego se instala el JDK 6, este programa se encuentran dentro del directorio creado por el instalador.

A.2 Ejecutando el programa.

Para ejecutar el programa se debe realizar un doble clic al icono con el nombre JGrafep, ubicado en el escritorio del PC, o entrando en la carpeta C:\Archivo de Programa\JGrafep\Dist y haciendo doble clic en el archivo JGrafep.jar, en caso de ejecutarlo en el sistema operativo Windows. Para otros sistemas operativos consulte con soporte técnico.

A.3 Abriendo la sesión del usuario.

Al ejecutar el programa JGrafep se mostrara la ventana principal, la cual tiene el siguiente aspecto:

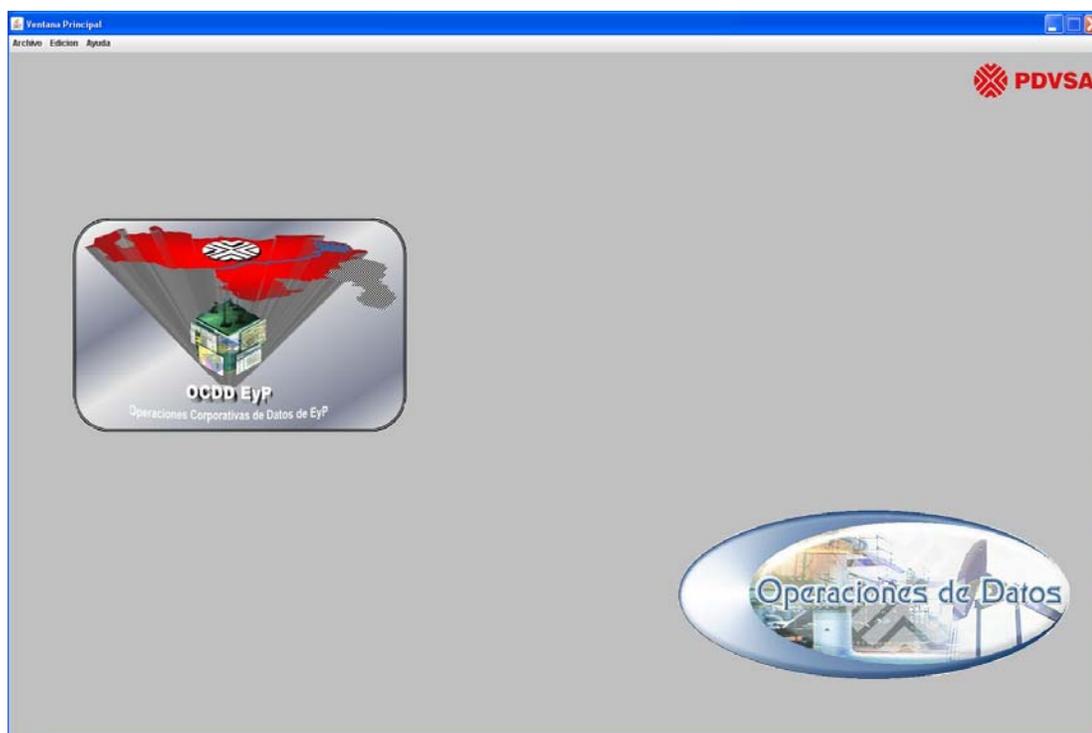


Figura A.1 Ventana Principal.

El usuario cuenta con las siguientes opciones en el menú del programa:

- *Archivo*
 - *Cargar Archivo de Datos*
 - *Cargar Archivo de Imagen*
 - *Graficar*
 - *Salir*
- *Edición*
 - *Editar Archivo de Datos*

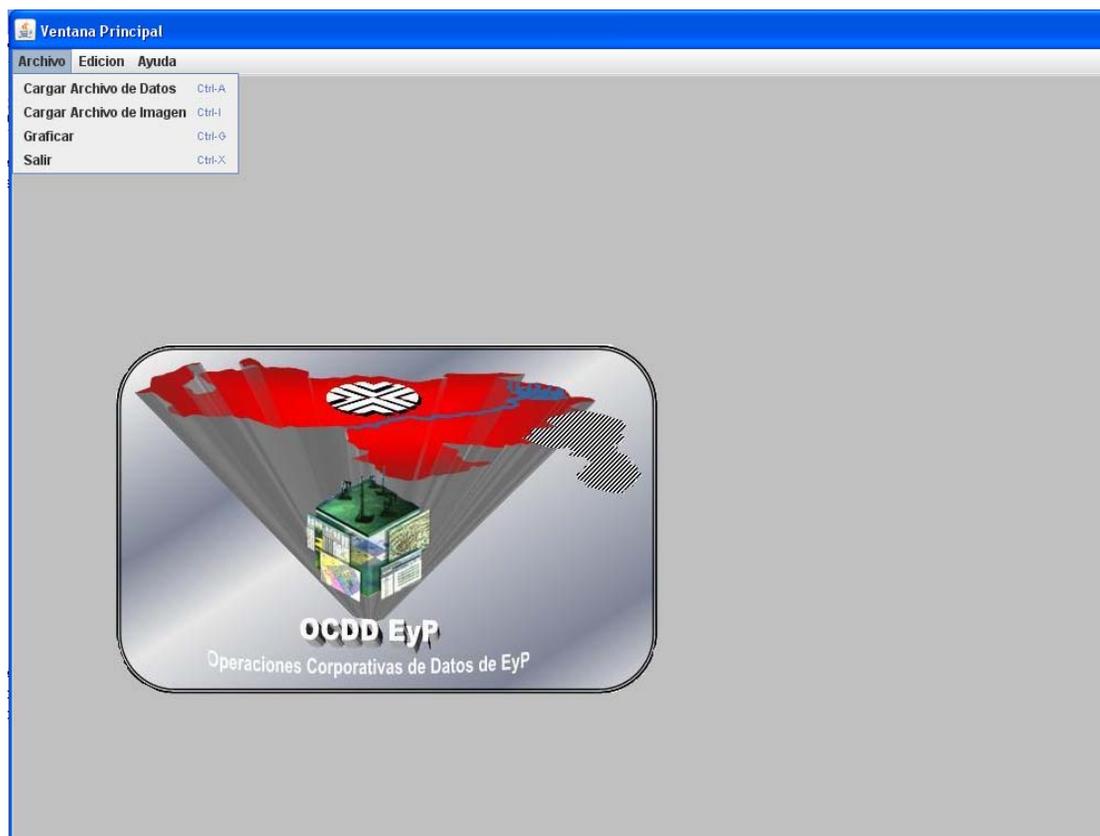


Figura A.2 Opciones del menú Archivo.

En el menú Archivo se cuentan con las siguientes opciones:

La opción *Cargar Archivo de Datos* despliega una ventana que le permite al usuario buscar un archivo de datos (Archivo LAS) dentro del computador y seleccionarlo para su apertura.

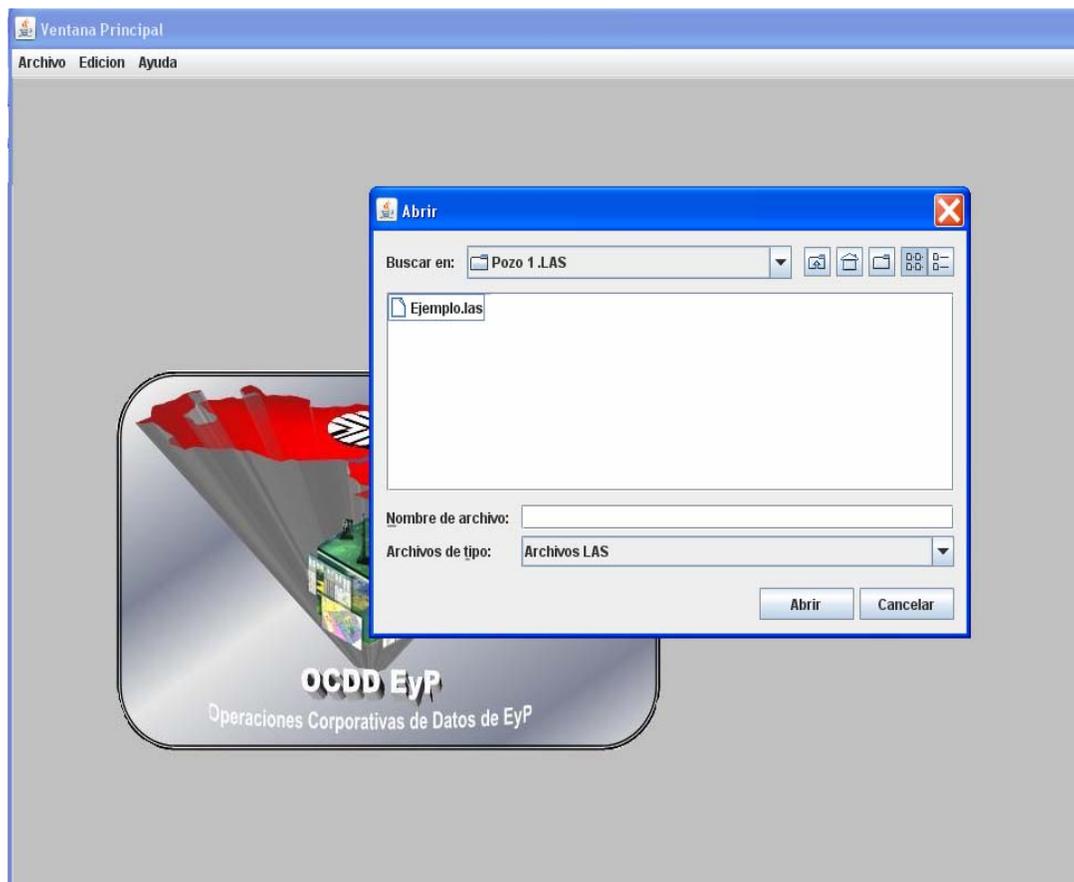


Figura A.3 Selector de Archivo de datos.

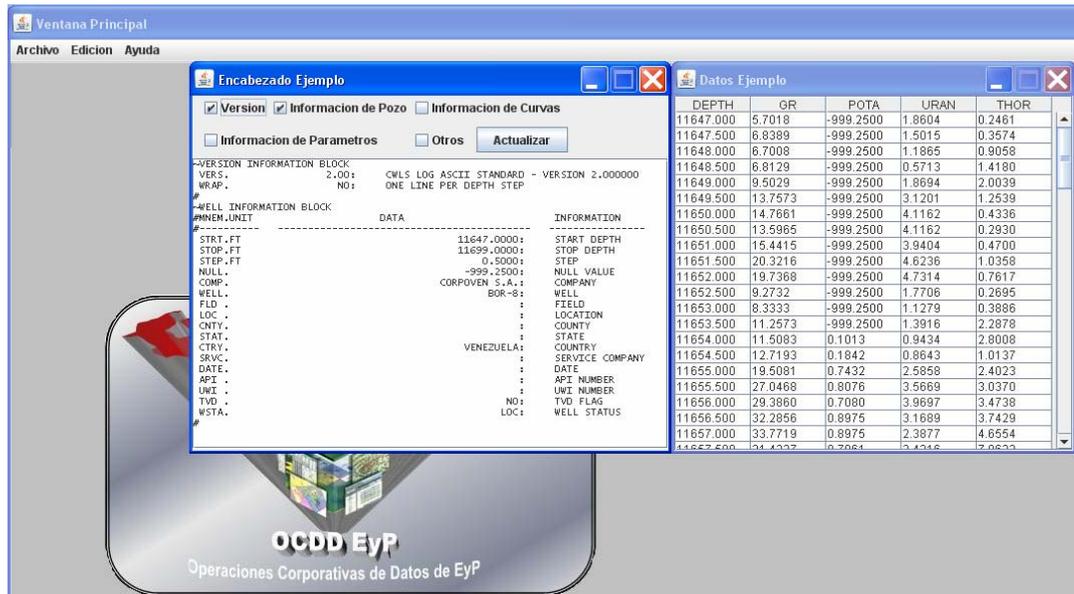


Figura A.4 Ventanas con los datos del archivo

La opción *Cargar Archivo de Imagen* despliega una ventana que le permite al usuario buscar un archivo de Imagen (Archivo JPG, TIF, PDF) dentro del computador y seleccionarlo para su apertura.

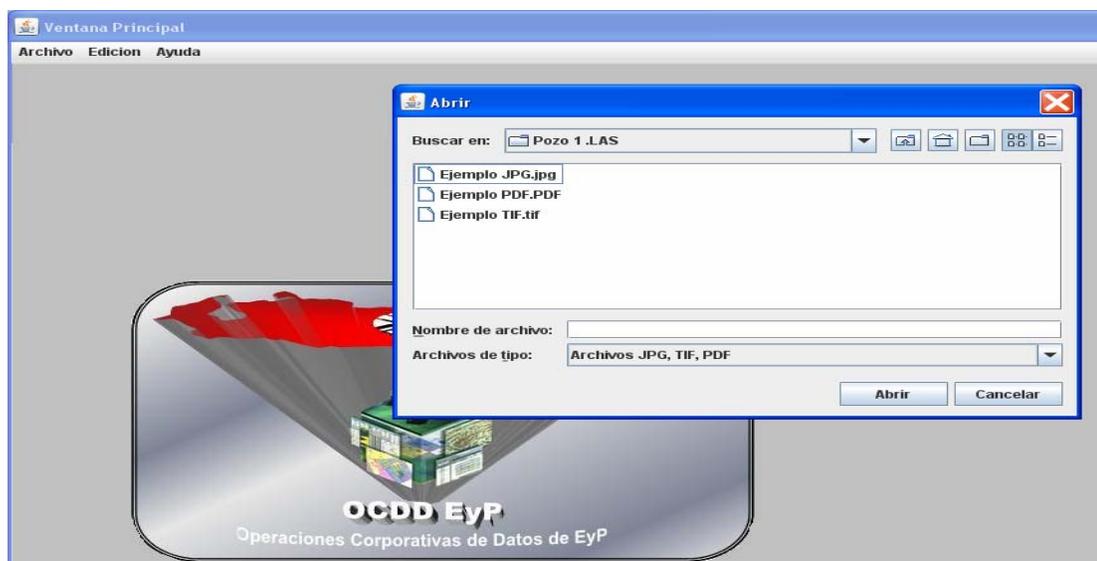


Figura A.5 Selector de Archivo de Imagen

La opción *Graficar* despliega una ventana que le permite al usuario buscar un archivo de datos (Archivo LAS) dentro del computador y seleccionarlo para su apertura y posterior Graficación.

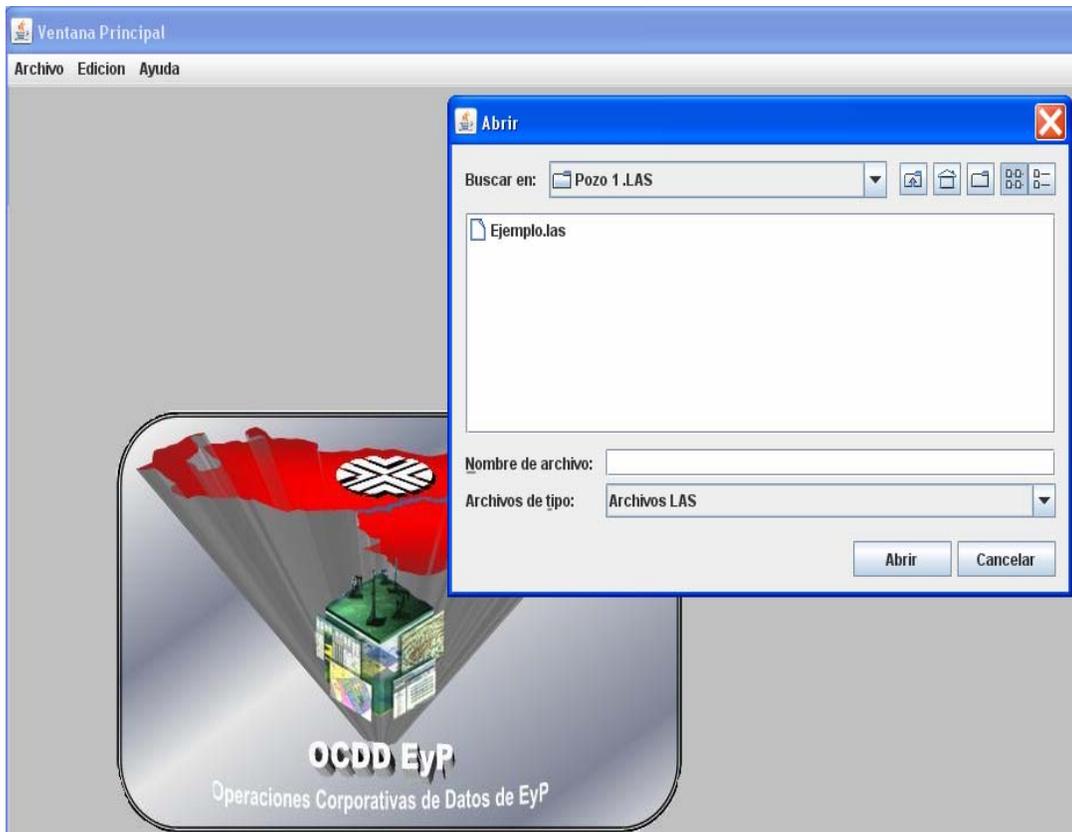


Figura A.6 Selector de Archivo de datos en la opción Graficar

Luego de escoger el archivo, se desplegará una ventana donde el usuario seleccionará los estudios que desee graficar, los cuales serán en el siguiente orden:

- 1.- El usuario seleccionará la escala de la gráfica a realizar.

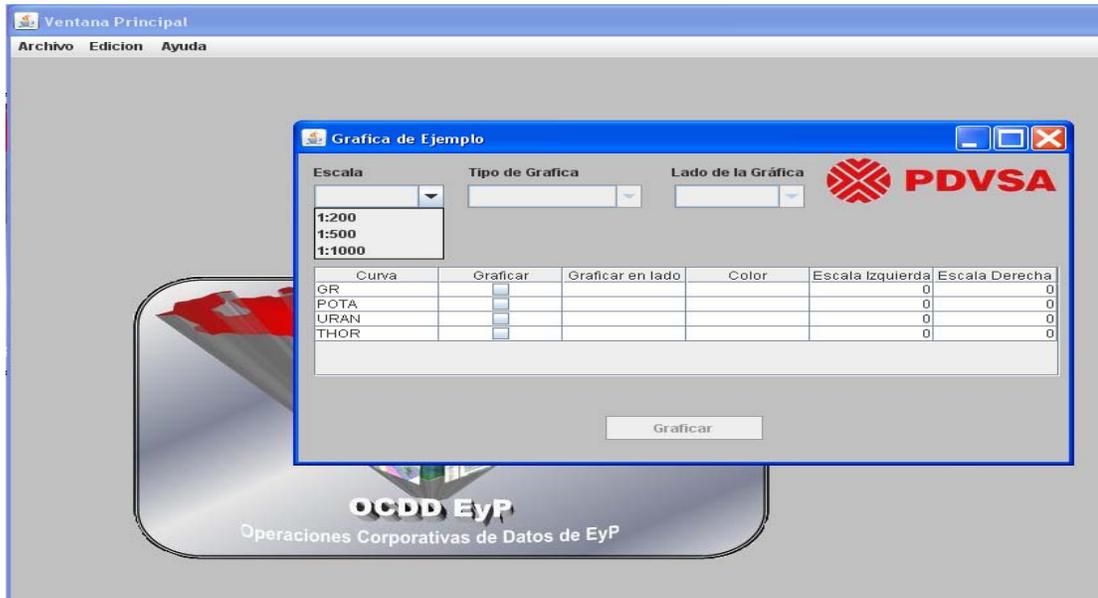


Figura A.7 Selección de Escala en ventana de Graficación

2.- El usuario seleccionara el tipo de grafica, entre las opciones tenemos Lineal-Lineal y Lineal-Logarítmica.

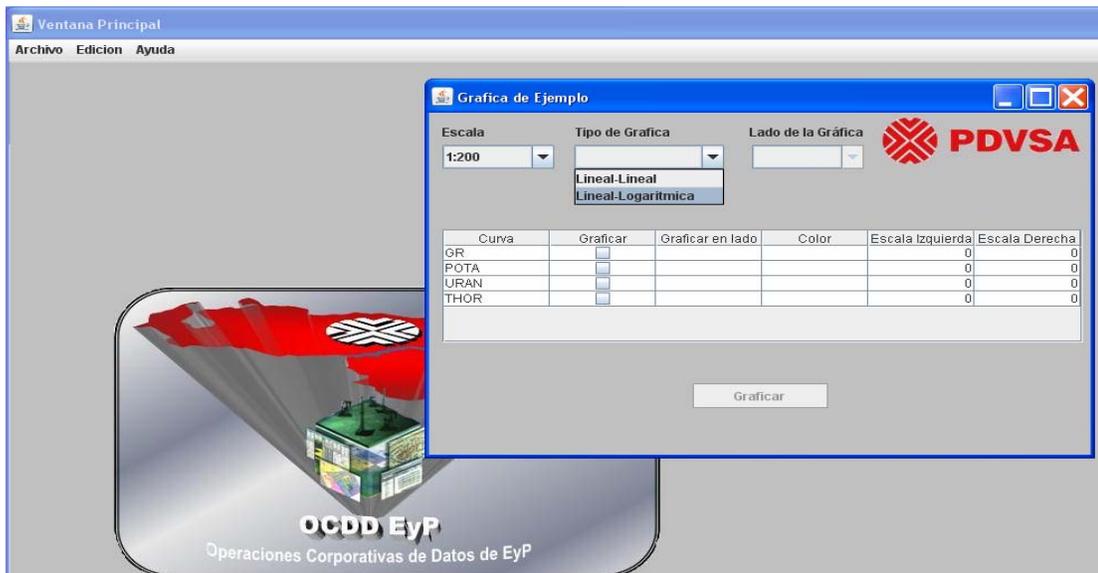


Figura A.8 Selección de Tipo de Grafica en ventana de Graficación

3.- El usuario hará la selección de las curvas a graficar.

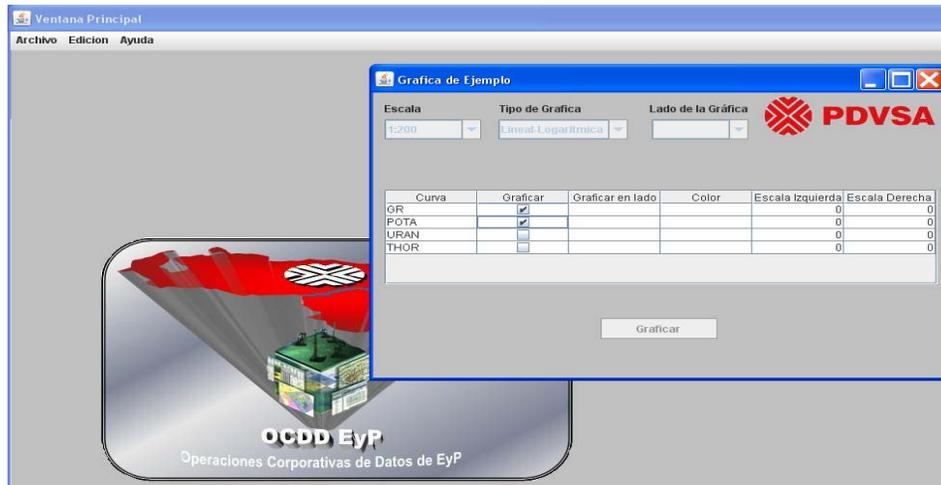


Figura A.9 Selección de las Curvas en ventana de Graficación

4.- Se selecciona el lado donde se va a graficar cada curva, primero haciendo clic en el recuadro siguiente al de la selección de la curva, para que se active el campo de selección de Lado de la Gráfica.

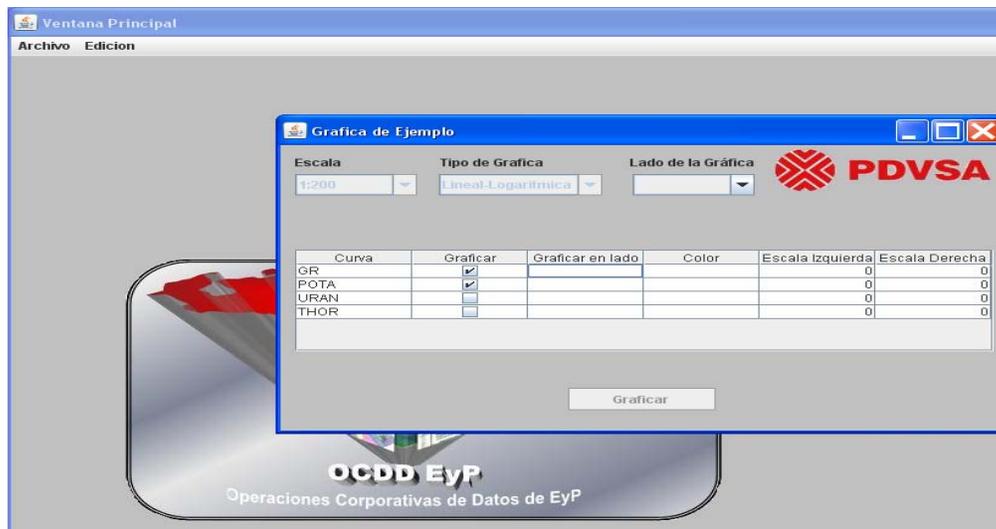


Figura A.10 Selección de la casilla en la cual se colocara el lado de la grafica.

Si el usuario selecciono Lineal-Logarítmico, en el campo de selección de Lado de la Grafica aparecerán las opciones Lineal y Logarítmico.

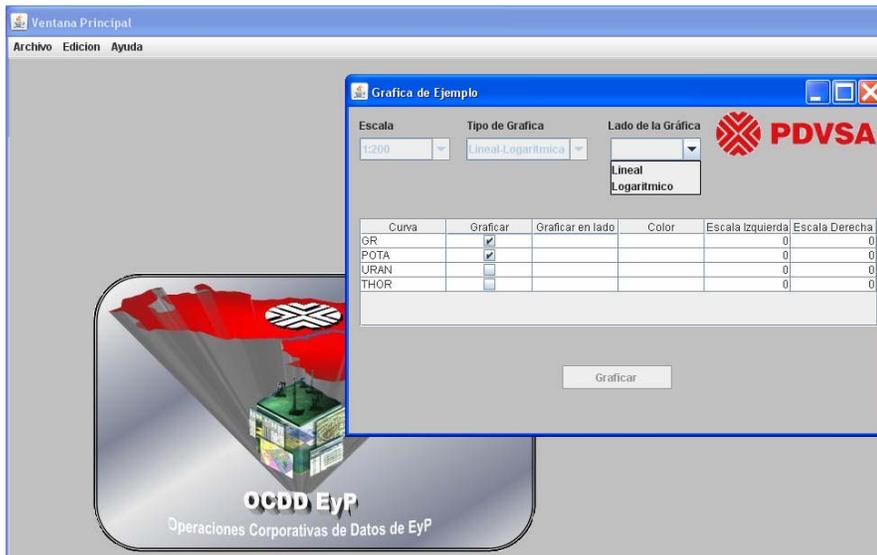


Figura A.11 Opción Lineal-Logarítmica.

Si el usuario eligió Lineal-Lineal en el campo Tipo de Grafica, en el campo de selección de Lado de la Grafica aparecerán las opciones Derecha e Izquierda.

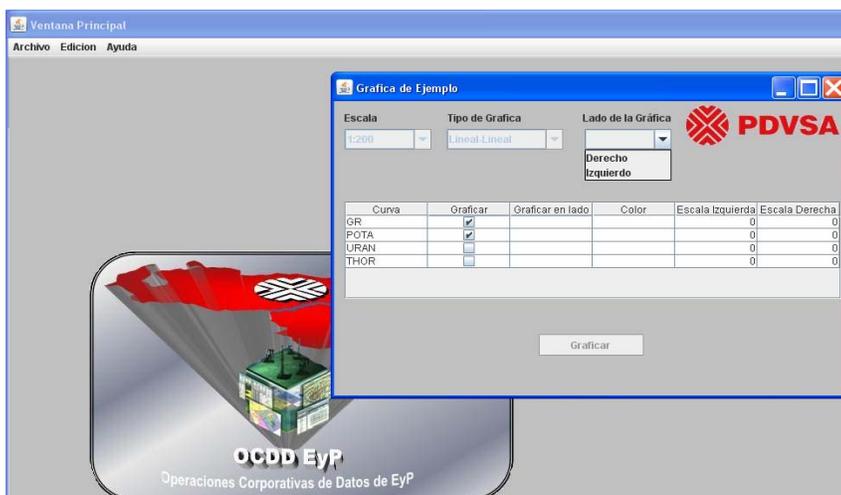


Figura A.12 Opción Lineal-Lineal.

5.- Se selecciona el color que tendrá cada curva, haciendo clic en el recuadro de la columna que tiene por nombre Color siguiente, para que se muestre por pantalla una ventana de selección de Color.

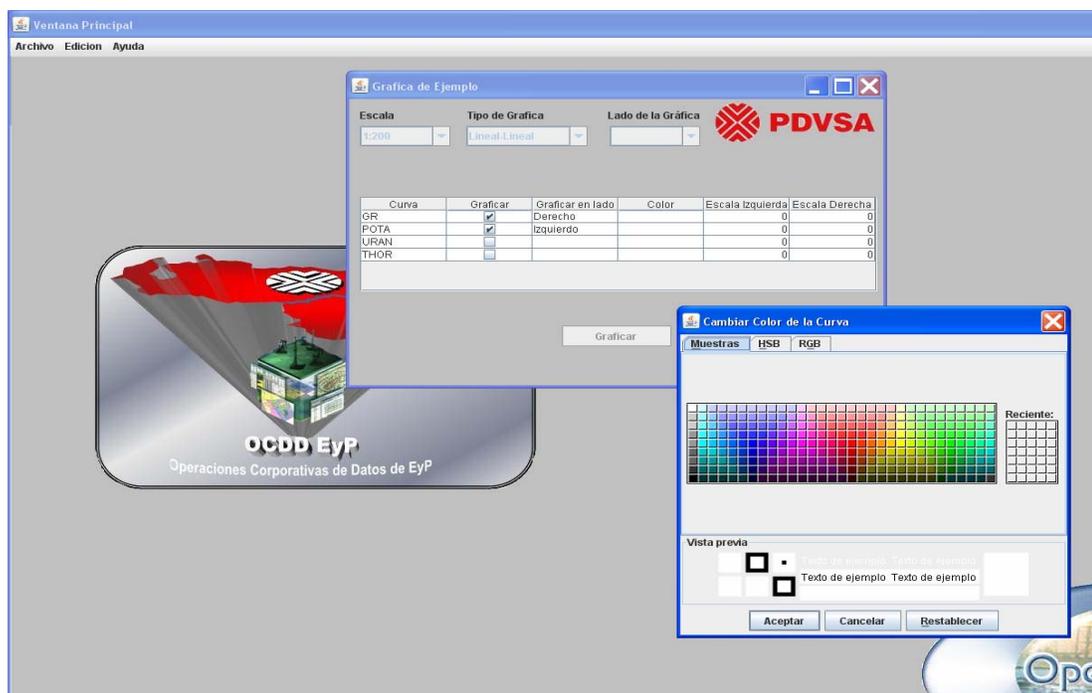


Figura A.13 Selección de Color

6.- Se procede a la introducción de las escalas de graficación, que son los rangos entre los cuales se va a graficar cada curva. Estos valores se ingresan manualmente, siempre y cuando las graficas sean lineales, ya que las graficas logarítmicas tienen por defecto asignados los valores 0,2 en el rango izquierdo y 2000 en el derecho.

Luego de haber seleccionado los rangos el botón Graficar se activara para poder realizar la grafica.

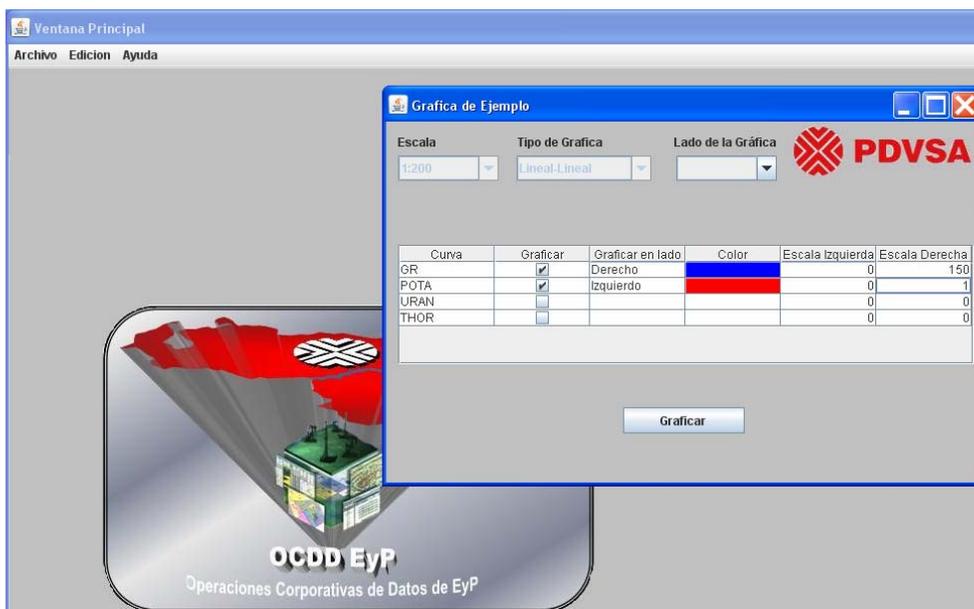


Figura A.14 Introducción de los valores de los rangos.

7.- Se pulsa el botón Graficar.

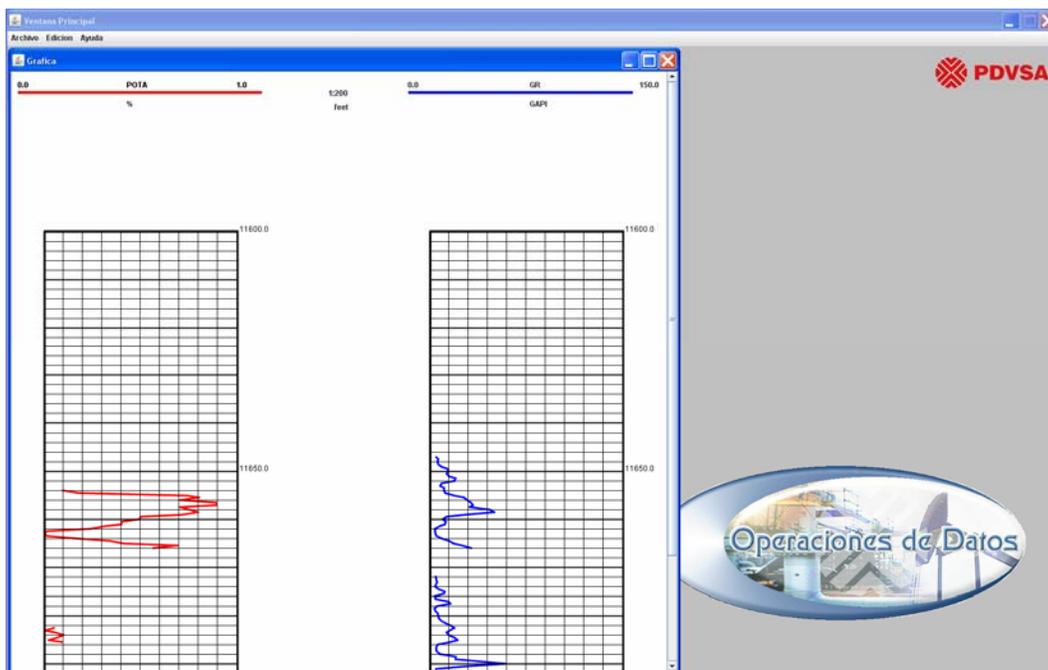


Figura A.15 Grafica lista.

En el menú Edición se cuentan con las siguientes opciones:

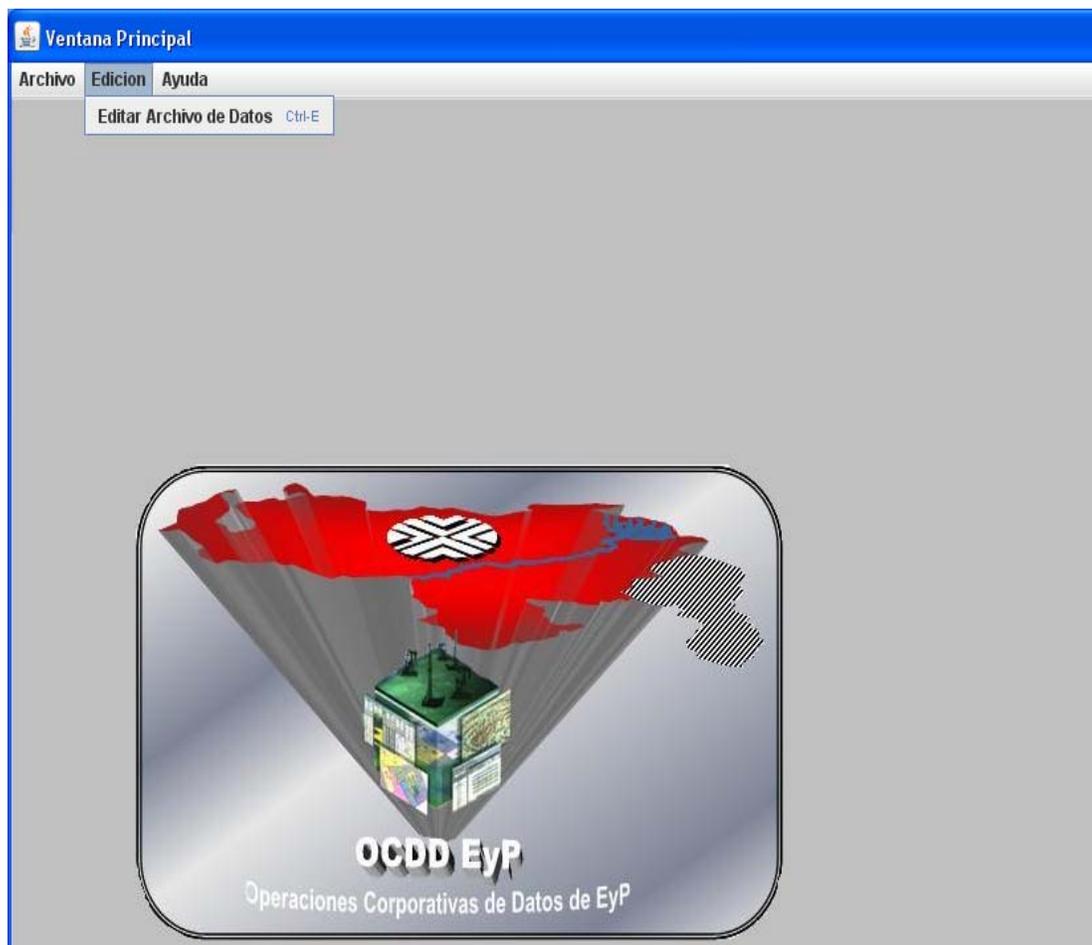


Figura A.16 Menú de Edición.

La opción *Editar Archivo de Datos* despliega una ventana que le permite al usuario editar los datos pertenecientes a las curvas en los archivos LAS, partiendo desde un archivo Origen, donde se obtienen los datos, y llevándolos a un archivo Final en el cual se guardarán todos los cambios con el nombre de archivo que se requiera.

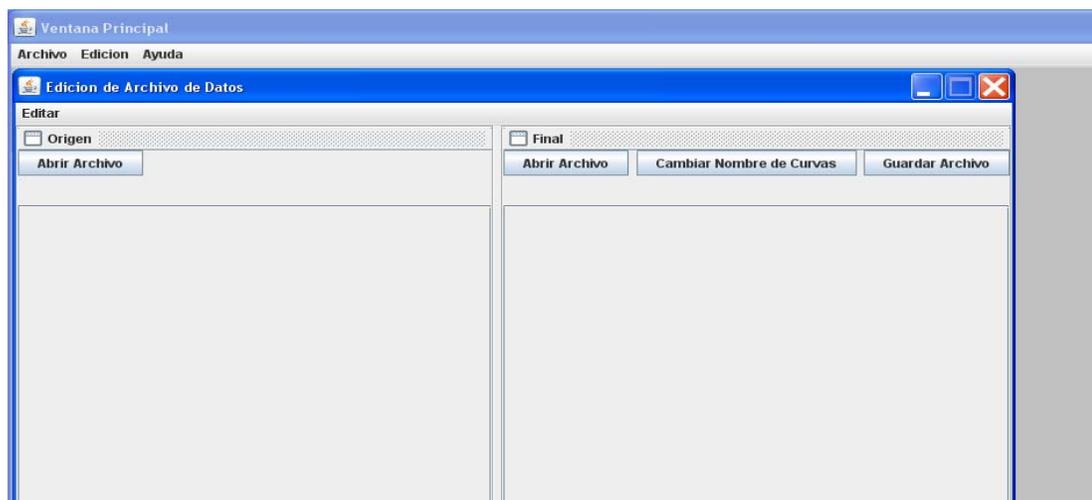


Figura A.17 Ventana de Edición de Archivo de Datos.

Para realizar el proceso de edición de un archivo LAS se deben seguir los siguientes pasos:

1.- Se debe de hacer clic en el botón Abrir Archivo de la ventana interna que tiene por nombre Origen, esta acción despliega una ventana que le permite al usuario buscar un archivo de datos (Archivo LAS) dentro del computador y seleccionarlo para su apertura.

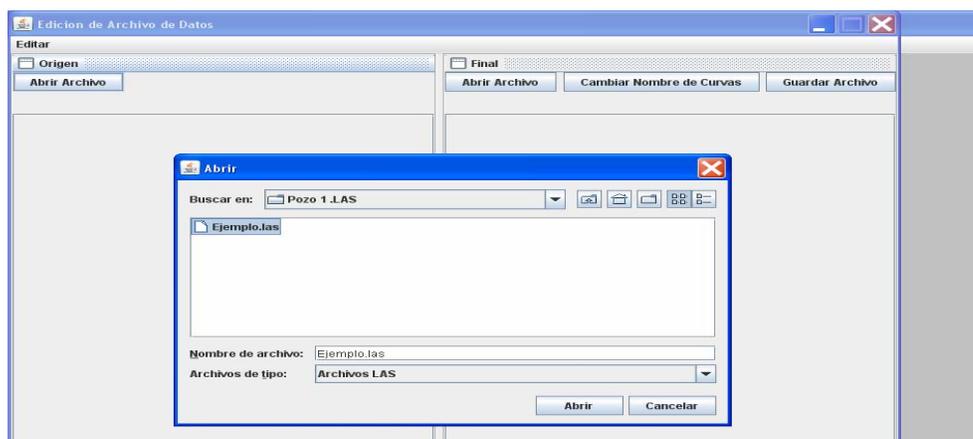


Figura A.18 Selector de Archivo en ventana Origen.

Luego de escoger el archivo, los datos pertenecientes a ese archivo serán mostrados en la ventana Origen, debajo del botón de Abrir Archivo se muestra el nombre del archivo con el que se esta trabajando.

| DEPTH | GR | POTA | URAN | THOR |
|-----------|---------|-----------|--------|--------|
| 11647.000 | 5.7018 | -999.2500 | 1.8604 | 0.2461 |
| 11647.500 | 6.8389 | -999.2500 | 1.5015 | 0.3574 |
| 11648.000 | 6.7008 | -999.2500 | 1.1865 | 0.9058 |
| 11648.500 | 6.8129 | -999.2500 | 0.5713 | 1.4180 |
| 11649.000 | 9.5029 | -999.2500 | 1.8694 | 2.0039 |
| 11649.500 | 13.7573 | -999.2500 | 3.1201 | 1.2539 |
| 11650.000 | 14.7661 | -999.2500 | 4.1162 | 0.4336 |
| 11650.500 | 13.5965 | -999.2500 | 4.1162 | 0.2930 |
| 11651.000 | 15.4415 | -999.2500 | 3.9404 | 0.4700 |
| 11651.500 | 20.3216 | -999.2500 | 4.6236 | 1.0358 |
| 11652.000 | 19.7368 | -999.2500 | 4.7314 | 0.7617 |
| 11652.500 | 9.2732 | -999.2500 | 1.7706 | 0.2695 |
| 11653.000 | 8.3333 | -999.2500 | 1.1279 | 0.3886 |
| 11653.500 | 11.2573 | -999.2500 | 1.3916 | 2.2878 |
| 11654.000 | 11.5083 | 0.1013 | 0.9434 | 2.8008 |
| 11654.500 | 12.7193 | 0.1842 | 0.8643 | 1.0137 |
| 11655.000 | 19.5081 | 0.7432 | 2.5858 | 2.4023 |
| 11655.500 | 27.0468 | 0.8076 | 3.5669 | 3.0370 |
| 11656.000 | 29.3860 | 0.7080 | 3.9697 | 3.4738 |
| 11656.500 | 32.2856 | 0.8975 | 3.1689 | 3.7429 |
| 11657.000 | 33.7719 | 0.8975 | 2.3877 | 4.6554 |
| 11657.500 | 31.4327 | 0.7061 | 2.4316 | 7.8633 |
| 11658.000 | 46.8993 | 0.7576 | 2.9671 | 9.3633 |
| 11658.500 | 50.3574 | 0.7992 | 3.8896 | 8.2375 |
| 11659.000 | 27.3392 | 0.7158 | 4.9658 | 4.9336 |
| 11659.500 | 13.7477 | 0.6038 | 4.1296 | 2.1334 |

Figura A.19 Datos en la Ventana Origen.

2.- Se debe de hacer clic en el botón Abrir Archivo de la ventana interna que tiene por nombre Final, esta acción despliega una ventana que le permite al usuario buscar un archivo de datos (Archivo LAS) dentro del computador y seleccionarlo para su apertura.

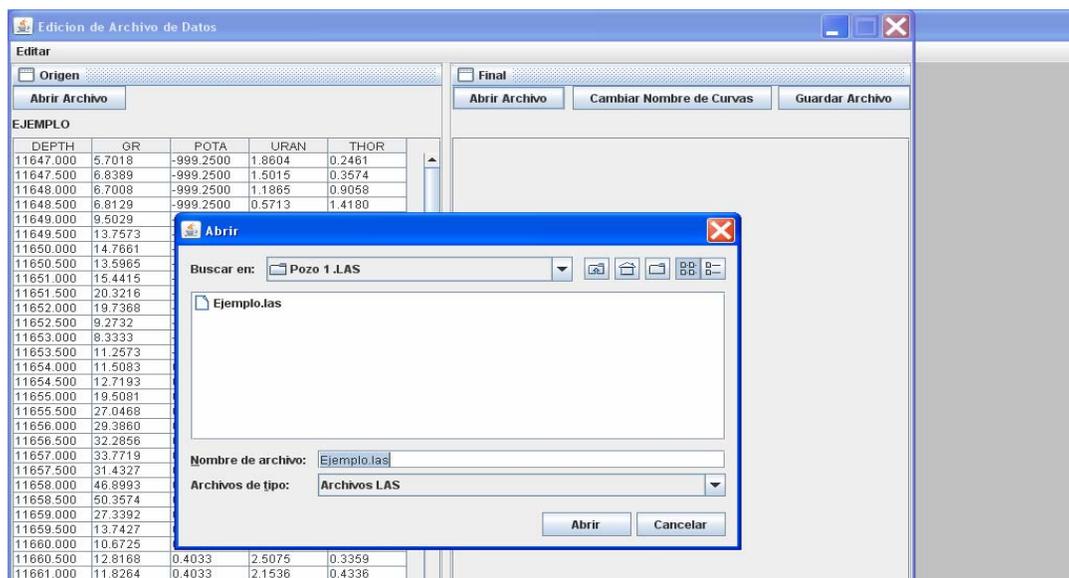


Figura A.20 Selector de Archivo en Ventana Final.

Luego de escoger el archivo, los datos pertenecientes a ese archivo serán mostrados en la ventana Final, debajo del botón de Abrir Archivo se muestra el nombre del archivo con el que se está trabajando.

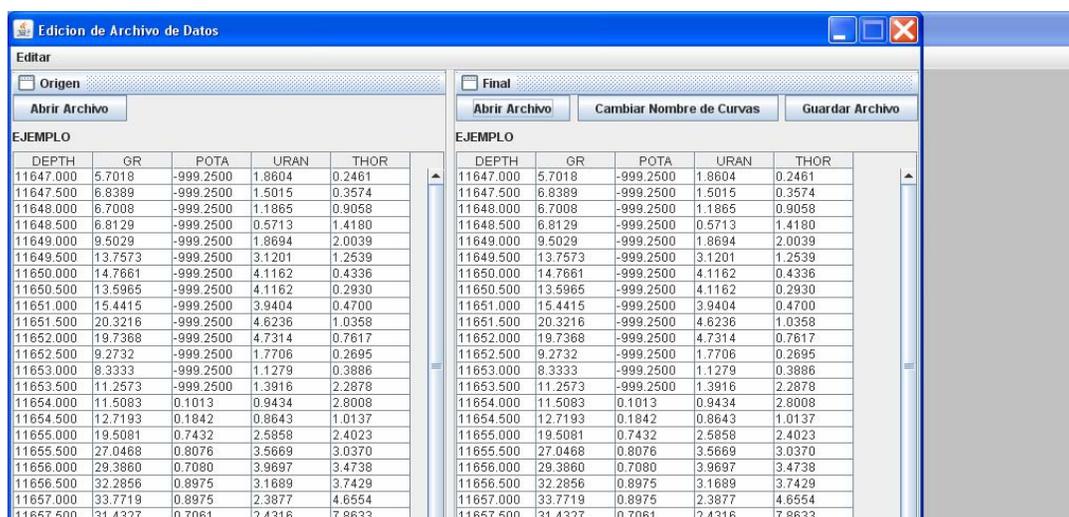


Figura A.21 Datos en la Ventana Final.

3.- Se procede a hacer la selección de las filas y las columnas a copiar desde la ventana Origen.

The screenshot displays the 'Edición de Archivo de Datos' window with two panes: 'Origen' and 'Final'. Both panes contain a table with the following columns: DEPTH, GR, POTA, URAN, and THOR. The 'Origen' table shows a sequence of data points for depths from 11647.000 to 11671.500. The 'Final' table shows the same data points, but many cells are replaced with '-999.2500', suggesting a selection or copying operation.

| DEPTH | GR | POTA | URAN | THOR |
|-----------|-----------|-----------|-----------|-----------|
| 11647.000 | 5.7018 | -999.2500 | 1.8604 | 0.2461 |
| 11647.500 | 6.8389 | -999.2500 | 1.5015 | 0.3574 |
| 11648.000 | 6.7008 | -999.2500 | 1.1865 | 0.9058 |
| 11648.500 | 6.8129 | -999.2500 | 0.5713 | 1.4180 |
| 11649.000 | 9.5029 | -999.2500 | 1.8694 | 2.0039 |
| 11649.500 | 13.7573 | -999.2500 | 3.1201 | 1.2539 |
| 11650.000 | 14.7661 | -999.2500 | 4.1162 | 0.4336 |
| 11650.500 | 13.5965 | -999.2500 | 4.1162 | 0.2930 |
| 11651.000 | 15.4415 | -999.2500 | 3.9404 | 0.4700 |
| 11651.500 | 20.3216 | -999.2500 | 4.6236 | 1.0358 |
| 11652.000 | 19.7368 | -999.2500 | 4.7314 | 0.7617 |
| 11652.500 | 9.2732 | -999.2500 | 1.7706 | 0.2695 |
| 11653.000 | 8.3333 | -999.2500 | 1.1279 | 0.3886 |
| 11653.500 | 11.2573 | -999.2500 | 1.3916 | 2.2878 |
| 11654.000 | 11.5083 | 0.1013 | 0.9434 | 2.8008 |
| 11654.500 | 12.7193 | 0.1842 | 0.8643 | 1.0137 |
| 11655.000 | 19.5081 | 0.7432 | 2.5858 | 2.4023 |
| 11655.500 | 27.0468 | 0.8076 | 3.5669 | 3.0370 |
| 11656.000 | 29.3860 | 0.7080 | 3.9697 | 3.4738 |
| 11656.500 | 32.2856 | 0.8975 | 3.1689 | 3.7429 |
| 11657.000 | 33.7719 | 0.8975 | 2.3877 | 4.6554 |
| 11657.500 | 31.4327 | 0.7061 | 2.4316 | 7.8633 |
| 11658.000 | 46.8993 | 0.7576 | 2.9671 | 9.3633 |
| 11658.500 | 50.3574 | 0.7992 | 3.8896 | 8.2375 |
| 11659.000 | 27.3392 | 0.7158 | 4.9658 | 4.9336 |
| 11659.500 | 13.7427 | 0.5038 | 4.1295 | 2.1334 |
| 11660.000 | 10.6725 | 0.4971 | 2.3584 | 0.8562 |
| 11660.500 | 12.8168 | 0.4033 | 2.5075 | 0.3359 |
| 11661.000 | 11.8264 | 0.4033 | 2.1536 | 0.4336 |
| 11661.500 | 7.9306 | 0.3154 | 1.2579 | 1.2539 |
| 11662.000 | 4.6909 | 0.2403 | 0.8053 | 0.5977 |
| 11662.500 | 4.2823 | 0.0069 | 0.4707 | 0.9459 |
| 11663.000 | 5.2314 | 0.0087 | 0.3304 | 2.3320 |
| 11663.500 | 5.5994 | 0.0105 | 0.3406 | 1.1388 |
| 11664.000 | 9.4482 | 0.1679 | 1.4962 | 0.9023 |
| 11664.500 | 15.3834 | 0.3460 | 2.5635 | 2.0073 |
| 11665.000 | 19.0476 | 0.4161 | 2.6514 | 3.1179 |
| 11665.500 | 25.3218 | 0.6982 | 0.3149 | 4.1602 |
| 11666.000 | 32.4585 | 0.5713 | 0.6597 | 3.4538 |
| 11666.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |

Figura A.22 Selección de Datos en la Ventana Origen.

4.- Ya hecha la selección se debe hacer clic en el botón Copiar que esta en la pestaña Editar.

The screenshot shows a software window titled "Edición de Archivo de Datos". It has a menu bar with "Editar" and a toolbar with "Copiar" (highlighted), "Pegar", "Final", "Abrir Archivo", "Cambiar Nombre de Curvas", and "Guardar Archivo". Below the toolbar are two data tables, both labeled "EJEMPLO".

| DEPTH | GR | POTA | URAN | THOR |
|-----------|-----------|-----------|-----------|-----------|
| 11647.000 | 5.7018 | -999.2500 | 1.8604 | 0.2461 |
| 11647.500 | 6.8389 | -999.2500 | 1.5015 | 0.3574 |
| 11648.000 | 6.7008 | -999.2500 | 1.1865 | 0.9058 |
| 11648.500 | 6.8129 | -999.2500 | 0.5713 | 1.4180 |
| 11649.000 | 9.5029 | -999.2500 | 1.8694 | 2.0039 |
| 11649.500 | 13.7573 | -999.2500 | 3.1201 | 1.2539 |
| 11650.000 | 14.7661 | -999.2500 | 4.1162 | 0.4336 |
| 11650.500 | 13.5965 | -999.2500 | 4.1162 | 0.2930 |
| 11651.000 | 15.4415 | -999.2500 | 3.9404 | 0.4700 |
| 11651.500 | 20.3216 | -999.2500 | 4.6236 | 1.0358 |
| 11652.000 | 19.7368 | -999.2500 | 4.7314 | 0.7617 |
| 11652.500 | 9.2732 | -999.2500 | 1.7706 | 0.2695 |
| 11653.000 | 8.3333 | -999.2500 | 1.1279 | 0.3886 |
| 11653.500 | 11.2573 | -999.2500 | 1.3916 | 2.2878 |
| 11654.000 | 11.5083 | 0.1013 | 0.9434 | 2.8008 |
| 11654.500 | 12.7193 | 0.1842 | 0.8643 | 1.0137 |
| 11655.000 | 19.5081 | 0.7432 | 2.5858 | 2.4023 |
| 11655.500 | 27.0468 | 0.8076 | 3.5669 | 3.0370 |
| 11656.000 | 29.3860 | 0.7080 | 3.9697 | 3.4738 |
| 11656.500 | 32.2856 | 0.8975 | 3.1689 | 3.7429 |
| 11657.000 | 33.7719 | 0.8975 | 2.3877 | 4.6554 |
| 11657.500 | 31.4327 | 0.7061 | 2.4316 | 7.8633 |
| 11658.000 | 46.8993 | 0.7576 | 2.9671 | 9.3633 |
| 11658.500 | 50.3574 | 0.7992 | 3.8896 | 8.2375 |
| 11659.000 | 27.3392 | 0.7158 | 4.9658 | 4.9336 |
| 11659.500 | 13.7427 | 0.5038 | 4.1295 | 2.1334 |
| 11660.000 | 10.6725 | 0.4971 | 2.3584 | 0.8562 |
| 11660.500 | 12.8168 | 0.4033 | 2.5075 | 0.3359 |
| 11661.000 | 11.8264 | 0.4033 | 2.1536 | 0.4336 |
| 11661.500 | 7.9306 | 0.3154 | 1.2579 | 1.2539 |
| 11662.000 | 4.6909 | 0.2403 | 0.8053 | 0.5977 |
| 11662.500 | 4.2823 | 0.0069 | 0.4707 | 0.9459 |
| 11663.000 | 5.2314 | 0.0087 | 0.3304 | 2.3320 |
| 11663.500 | 5.5994 | 0.0105 | 0.3406 | 1.1388 |
| 11664.000 | 9.4482 | 0.1679 | 1.4962 | 0.9023 |
| 11664.500 | 15.3834 | 0.3460 | 2.5635 | 2.0073 |
| 11665.000 | 19.0476 | 0.4161 | 2.6514 | 3.1179 |
| 11665.500 | 25.3218 | 0.6982 | 0.3149 | 4.1602 |
| 11666.000 | 32.4585 | 0.5713 | 0.6597 | 3.4538 |
| 11666.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |

Figura A.23 Copiado de la Selección de Datos en la Ventana Origen.

5.- Se continua haciendo la selección de el recuadro en la ventana Final donde se van a colocar los datos seleccionados en la ventana Origen.

The screenshot shows the 'Edición de Archivo de Datos' application. The interface is divided into two main sections: 'Origen' (Origin) and 'Final' (Final). Both sections contain a table of data points with the following columns: DEPTH, GR, POTA, URAN, and THOR. The 'Origen' table lists data points from 11647.000 to 11671.500. The 'Final' table lists the same data points, but the row for '11655.000' is highlighted in blue, indicating it has been selected for transfer. The 'Final' section also includes buttons for 'Abrir Archivo', 'Cambiar Nombre de Curvas', and 'Guardar Archivo'.

| DEPTH | GR | POTA | URAN | THOR |
|-----------|-----------|-----------|-----------|-----------|
| 11647.000 | 5.7018 | -999.2500 | 1.8604 | 0.2461 |
| 11647.500 | 6.8389 | -999.2500 | 1.5015 | 0.3574 |
| 11648.000 | 6.7008 | -999.2500 | 1.1865 | 0.9058 |
| 11648.500 | 6.8129 | -999.2500 | 0.5713 | 1.4180 |
| 11649.000 | 9.5029 | -999.2500 | 1.8694 | 2.0039 |
| 11649.500 | 13.7573 | -999.2500 | 3.1201 | 1.2539 |
| 11650.000 | 14.7661 | -999.2500 | 4.1162 | 0.4336 |
| 11650.500 | 13.5965 | -999.2500 | 4.1162 | 0.2930 |
| 11651.000 | 15.4415 | -999.2500 | 3.9404 | 0.4700 |
| 11651.500 | 20.3216 | -999.2500 | 4.6236 | 1.0358 |
| 11652.000 | 19.7368 | -999.2500 | 4.7314 | 0.7617 |
| 11652.500 | 9.2732 | -999.2500 | 1.7706 | 0.2695 |
| 11653.000 | 8.3333 | -999.2500 | 1.1279 | 0.3886 |
| 11653.500 | 11.2573 | -999.2500 | 1.3916 | 2.2878 |
| 11654.000 | 11.5083 | 0.1013 | 0.9434 | 2.8008 |
| 11654.500 | 12.7193 | 0.1842 | 0.8643 | 1.0137 |
| 11655.000 | 19.5081 | 0.7432 | 2.5858 | 2.4023 |
| 11655.500 | 27.0468 | 0.8076 | 3.5669 | 3.0370 |
| 11656.000 | 29.3860 | 0.7080 | 3.9697 | 3.4738 |
| 11656.500 | 32.2856 | 0.8975 | 3.1689 | 3.7429 |
| 11657.000 | 33.7719 | 0.8975 | 2.3877 | 4.6554 |
| 11657.500 | 31.4327 | 0.7061 | 2.4316 | 7.8633 |
| 11658.000 | 46.8993 | 0.7576 | 2.9671 | 9.3633 |
| 11658.500 | 50.3574 | 0.7992 | 3.8896 | 8.2375 |
| 11659.000 | 27.3392 | 0.7158 | 4.9658 | 4.9336 |
| 11659.500 | 13.7427 | 0.5038 | 4.1295 | 2.1334 |
| 11660.000 | 10.6725 | 0.4971 | 2.3584 | 0.8562 |
| 11660.500 | 12.8168 | 0.4033 | 2.5075 | 0.3359 |
| 11661.000 | 11.8264 | 0.4033 | 2.1536 | 0.4336 |
| 11661.500 | 7.9306 | 0.3154 | 1.2579 | 1.2539 |
| 11662.000 | 4.6909 | 0.2403 | 0.8053 | 0.5977 |
| 11662.500 | 4.2823 | 0.0069 | 0.4707 | 0.9459 |
| 11663.000 | 5.2314 | 0.0087 | 0.3304 | 2.3320 |
| 11663.500 | 5.5994 | 0.0105 | 0.3406 | 1.1388 |
| 11664.000 | 9.4482 | 0.1679 | 1.4962 | 0.9023 |
| 11664.500 | 15.3834 | 0.3460 | 2.5635 | 2.0073 |
| 11665.000 | 19.0476 | 0.4161 | 2.6514 | 3.1179 |
| 11665.500 | 25.3218 | 0.6982 | 0.3149 | 4.1602 |
| 11666.000 | 32.4585 | 0.5713 | 0.6597 | 3.4538 |
| 11666.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |

Figura A.24 Selección de la ubicación de los Datos en la Ventana Final.

6.- Ya hecha la selección se debe hacer clic en el botón Pegar que esta en la pestaña Editar.

The screenshot shows a software window titled "Edición de Archivo de Datos" with two panes. The left pane, labeled "Origen", contains a table with 5 columns: DEPTH, GR, POTA, URAN, and THOR. The right pane, labeled "Final", contains an identical table. The "Final" pane also features three buttons: "Abrir Archivo", "Cambiar Nombre de Curvas", and "Guardar Archivo". The data in both tables is as follows:

| DEPTH | GR | POTA | URAN | THOR |
|-----------|-----------|-----------|-----------|-----------|
| 11647.000 | 5.7018 | -999.2500 | 1.8604 | 0.2461 |
| 11647.500 | 6.8389 | -999.2500 | 1.5015 | 0.3574 |
| 11648.000 | 6.7008 | -999.2500 | 1.1865 | 0.9058 |
| 11648.500 | 6.8129 | -999.2500 | 0.5713 | 1.4180 |
| 11649.000 | 9.5029 | -999.2500 | 1.8694 | 2.0039 |
| 11649.500 | 13.7573 | -999.2500 | 3.1201 | 1.2539 |
| 11650.000 | 14.7661 | -999.2500 | 4.1162 | 0.4336 |
| 11650.500 | 13.5965 | -999.2500 | 4.1162 | 0.2930 |
| 11651.000 | 15.4415 | -999.2500 | 3.9404 | 0.4700 |
| 11651.500 | 20.3216 | -999.2500 | 4.6236 | 1.0358 |
| 11652.000 | 19.7368 | -999.2500 | 4.7314 | 0.7617 |
| 11652.500 | 9.2732 | -999.2500 | 1.7706 | 0.2695 |
| 11653.000 | 8.3333 | -999.2500 | 1.1279 | 0.3886 |
| 11653.500 | 11.2573 | -999.2500 | 1.3916 | 2.2878 |
| 11654.000 | 11.5083 | 0.1013 | 0.9434 | 2.8008 |
| 11654.500 | 12.7193 | 0.1842 | 0.8643 | 1.0137 |
| 11655.000 | 19.5081 | 0.7432 | 2.5858 | 2.4023 |
| 11655.500 | 27.0468 | 0.8076 | 3.5669 | 3.0370 |
| 11656.000 | 29.3860 | 0.7080 | 3.9697 | 3.4738 |
| 11656.500 | 32.2856 | 0.8975 | 3.1689 | 3.7429 |
| 11657.000 | 33.7719 | 0.8975 | 2.3877 | 4.6554 |
| 11657.500 | 31.4327 | 0.7061 | 2.4316 | 7.8633 |
| 11658.000 | 46.8993 | 0.7576 | 2.9671 | 9.3633 |
| 11658.500 | 50.3574 | 0.7992 | 3.8896 | 8.2375 |
| 11659.000 | 27.3392 | 0.7158 | 4.9658 | 4.9336 |
| 11659.500 | 13.7427 | 0.5038 | 4.1295 | 2.1334 |
| 11660.000 | 10.6725 | 0.4971 | 2.3584 | 0.8562 |
| 11660.500 | 12.8168 | 0.4033 | 2.5075 | 0.3359 |
| 11661.000 | 11.8264 | 0.4033 | 2.1536 | 0.4336 |
| 11661.500 | 7.9306 | 0.3154 | 1.2579 | 1.2539 |
| 11662.000 | 4.6909 | 0.2403 | 0.8053 | 0.5977 |
| 11662.500 | 4.2823 | 0.0069 | 0.4707 | 0.9459 |
| 11663.000 | 5.2314 | 0.0087 | 0.3304 | 2.3320 |
| 11663.500 | 5.5994 | 0.0105 | 0.3406 | 1.1388 |
| 11664.000 | 9.4482 | 0.1679 | 1.4962 | 0.9023 |
| 11664.500 | 15.3834 | 0.3460 | 2.5635 | 2.0073 |
| 11665.000 | 19.0476 | 0.4161 | 2.6514 | 3.1179 |
| 11665.500 | 25.3218 | 0.6982 | 0.3149 | 4.1602 |
| 11666.000 | 32.4585 | 0.5713 | 0.6597 | 3.4538 |
| 11666.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |

Figura A.25 Pegado de la Selección de los Datos en la Ventana Final.

Si se requiere cambiar los nombres de los estudios antes de guardar los datos en un nuevo archivo, se debe hacer clic en el botón Cambiar Nombre de Curvas en la ventana Final, y se desplegará en pantalla una nueva ventana con los nombres de las curvas en el archivo. En la fila de color gris están los nombres de los estudios, y en la fila blanca están los nombres que se le van a asignar a los estudios, después de haber cambiado los nombres, se debe pulsar el botón Guardar.

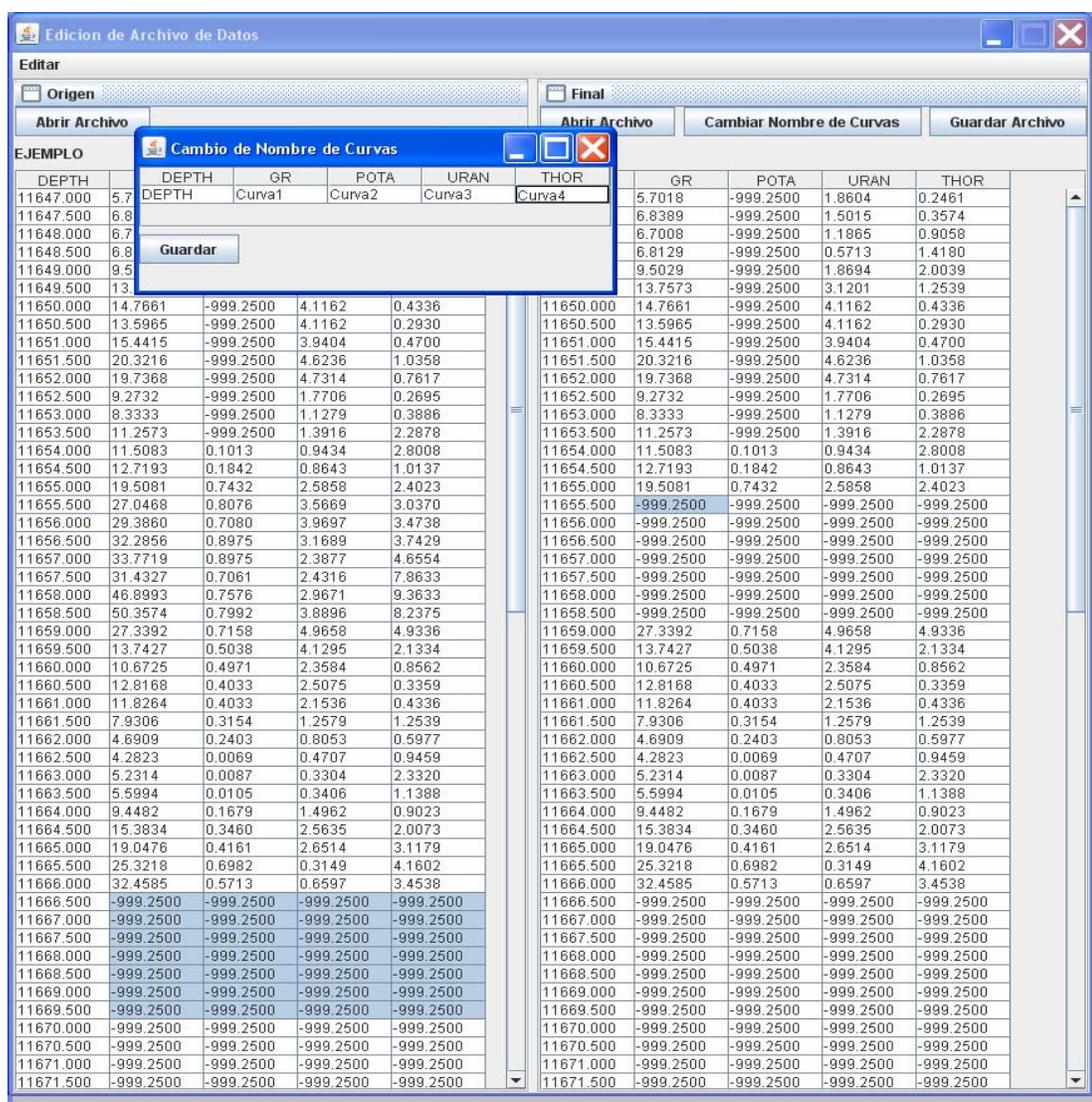


Figura A.26 Cambio de nombre de las Curvas en la Ventana Final.

Después de Guardar los cambios hechos en los nombres de las curvas, los datos a los cuales se le hicieron modificaciones en la ventana final desaparecerán, de este modo se recomienda hacer el cambio de nombre de las curvas al principio de las modificaciones.

| DEPTH | GR | POTA | URAN | THOR |
|-----------|-----------|-----------|-----------|-----------|
| 11647.000 | 5.7018 | -999.2500 | 1.8604 | 0.2461 |
| 11647.500 | 6.8389 | -999.2500 | 1.5015 | 0.3574 |
| 11648.000 | 6.7008 | -999.2500 | 1.1865 | 0.9058 |
| 11648.500 | 6.8129 | -999.2500 | 0.5713 | 1.4180 |
| 11649.000 | 9.5029 | -999.2500 | 1.8694 | 2.0039 |
| 11649.500 | 13.7573 | -999.2500 | 3.1201 | 1.2539 |
| 11650.000 | 14.7661 | -999.2500 | 4.1162 | 0.4336 |
| 11650.500 | 13.5965 | -999.2500 | 4.1162 | 0.2930 |
| 11651.000 | 15.4415 | -999.2500 | 3.9404 | 0.4700 |
| 11651.500 | 20.3216 | -999.2500 | 4.6236 | 1.0358 |
| 11652.000 | 19.7368 | -999.2500 | 4.7314 | 0.7617 |
| 11652.500 | 9.2732 | -999.2500 | 1.7706 | 0.2695 |
| 11653.000 | 8.3333 | -999.2500 | 1.1279 | 0.3886 |
| 11653.500 | 11.2573 | -999.2500 | 1.3916 | 2.2878 |
| 11654.000 | 11.5083 | 0.1013 | 0.9434 | 2.8008 |
| 11654.500 | 12.7193 | 0.1842 | 0.8643 | 1.0137 |
| 11655.000 | 19.5081 | 0.7432 | 2.5858 | 2.4023 |
| 11655.500 | 27.0468 | 0.8076 | 3.5669 | 3.0370 |
| 11656.000 | 29.3860 | 0.7080 | 3.9697 | 3.4738 |
| 11656.500 | 32.2856 | 0.8975 | 3.1689 | 3.7429 |
| 11657.000 | 33.7719 | 0.8975 | 2.3877 | 4.6554 |
| 11657.500 | 31.4327 | 0.7061 | 2.4316 | 7.8633 |
| 11658.000 | 46.8993 | 0.7576 | 2.9671 | 9.3633 |
| 11658.500 | 50.3574 | 0.7992 | 3.8896 | 8.2375 |
| 11659.000 | 27.3392 | 0.7158 | 4.9658 | 4.9336 |
| 11659.500 | 13.7427 | 0.5038 | 4.1295 | 2.1334 |
| 11660.000 | 10.6725 | 0.4971 | 2.3584 | 0.8562 |
| 11660.500 | 12.8168 | 0.4033 | 2.5075 | 0.3359 |
| 11661.000 | 11.8264 | 0.4033 | 2.1536 | 0.4336 |
| 11661.500 | 7.9306 | 0.3154 | 1.2579 | 1.2539 |
| 11662.000 | 4.6909 | 0.2403 | 0.8053 | 0.5977 |
| 11662.500 | 4.2823 | 0.0069 | 0.4707 | 0.9459 |
| 11663.000 | 5.2314 | 0.0087 | 0.3304 | 2.3320 |
| 11663.500 | 5.5994 | 0.0105 | 0.3406 | 1.1388 |
| 11664.000 | 9.4482 | 0.1679 | 1.4962 | 0.9023 |
| 11664.500 | 15.3834 | 0.3460 | 2.5635 | 2.0073 |
| 11665.000 | 19.0476 | 0.4161 | 2.6514 | 3.1179 |
| 11665.500 | 25.3218 | 0.6982 | 0.3149 | 4.1602 |
| 11666.000 | 32.4585 | 0.5713 | 0.6597 | 3.4538 |
| 11666.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11667.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11668.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11669.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11670.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.000 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |
| 11671.500 | -999.2500 | -999.2500 | -999.2500 | -999.2500 |

Figura A.27 Cambios realizados en la Ventana Final.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

| | |
|-----------|---|
| TÍTULO | “DESARROLLO DE UN SOFTWARE QUE PERMITA LA EDICION Y GRAFICACION DE INFORMACION RELACIONADA CON PERFILES DE POZO MANEJADA EN EL DEPARTAMENTO DE EXPLORACION Y PRODUCCION DE LA EMPRESA PDVSA PUERTO LA CRUZ” |
| SUBTÍTULO | |

AUTOR (ES):

| APELLIDOS Y NOMBRES | CÓDIGO CULAC / E MAIL |
|-------------------------------|---|
| Julio César Martínez González | CVLAC: 13783418 E MAIL: juliocmg0111@hotmail.com |
| | CVLAC: E MAIL: |

PALABRAS O FRASES CLAVES:

Archivos LAS _____Graficación _____Ciclo de vida del Software _____UML _____Casos de uso _____Diagrama de Secuencia _____Diagrama de clases _____Diagrama de Colaboración. _____

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

| ÀREA | SUBÀREA |
|---------------------------------|---------------------------|
| Ingeniería y Ciencias Aplicadas | Ingeniería en Computación |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

RESUMEN (ABSTRACT):

Este proyecto tiene como finalidad la realización de un software que permita la edición y graficación de información relacionada con perfiles de pozo manejada en el departamento de exploración y producción de la empresa PDVSA puerto la cruz. Con esto se plantea optimizar y reducir los tiempos y los errores, en la realización de las operaciones llevadas a cabo en la Gerencia.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

| APELLIDOS Y NOMBRES | ROL / CÓDIGO CVLAC / E_MAIL | | | | |
|---------------------|-----------------------------|-------------------------|---------|----|---------|
| Gabriela Veracierta | ROL | CA | AS X | TU | JU |
| | CVLAC: | 14.616.683 | | | |
| | E_MAIL | gveracierta@hotmail.com | | | |
| | E_MAIL | | | | |
| V́ctor Mújica | ROL | CA | AS | TU | JU X |
| | CVLAC: | 14.054.907 | | | |
| | E_MAIL | Victormujicayahoo.com | | | |
| | E_MAIL | | | | |
| Claudio Cort́nez | ROL | CA | AS | TU | JU X |
| | CVLAC: | 12.155.334 | | | |
| | E_MAIL | Cl_cortinez@cantv.net | | | |
| | E_MAIL | | | | |
| | ROL | CA | AS | TU | JU |
| | CVLAC: | | | | |
| | E_MAIL | | | | |
| | E_MAIL | | | | |

FECHA DE DISCUSIÓN Y APROBACIÓN:

| | | |
|-----|-----|-----|
| | | |
| AÑO | MES | DÍA |

LENGUAJE. SPA.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

| NOMBRE DE ARCHIVO | TIPO MIME |
|---|--------------------|
| TESIS.Graficación y edición de perfiles de pozo.doc | Application/msword |
| | |
| | |
| | |

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G
H I J K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u v w x
y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación.

NIVEL ASOCIADO CON EL TRABAJO

Pre-Grado.

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas.

INSTITUCIÓN:

Universidad de Oriente / Núcleo Anzoátegui.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

Artículo 44 “Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y solo podrán ser utilizadas a otros fines con el consentimiento del consejo de núcleo respectivo, quién lo participará al consejo universitario”

Julio C. Martínez G.

AUTOR

Ing. Gabriela Veracierta

TUTOR

Ing. Víctor Mújica

JURADO

Ing. Claudio Cortínez

JURADO

Ing. Luis Bastardo

CORDINADOR DE LA SUBCOMISION