

Fig. 1. Disposición del equipo para la adquisición de una traza sísmica.

2 Ondas Sísmicas

Las ondas sísmicas son señales analíticas que poseen una parte real y otra imaginaria, de la cual sólo la parte real es detectada y visualizada. Esto se corresponde con un fasor dependiente del tiempo, de esta forma se puede analizar a la traza sísmica $g(t)$ como:

$$g(t) = R(t) \cos \theta(t) \quad (1)$$

donde $R(t)$ es la envolvente de la traza sísmica y $\theta(t)$ es su fase. Para describir la traza sísmica, podría imaginarse un vector perpendicular al eje del tiempo cuya longitud varía en el tiempo y que rota alrededor de dicho eje como una función temporal. La proyección de este vector rotatorio sobre el plano real genera la traza sísmica convencional $g(t)$, pero si es proyectado sobre un plano imaginario se obtiene la traza cuadratura $h(t)$, la cual se define así:

$$h(t) = R(t) \sin \theta(t) \quad (2)$$

Una onda sísmica perturba las partículas del medio a través de la cual se propaga alternando su posición de equilibrio y por eso se puede decir que la traza sísmica convencional $g(t)$ puede ser vista como una medida de la energía cinética. La energía al viajar se refleja y se refracta cuando encuentra el límite entre dos materiales con propiedades acústicas diferentes, los geófonos colocados en la superficie miden la energía acústica reflejada, convirtiéndola en una señal eléctrica que se muestra como una traza sísmica, la cual es analizada en términos de Amplitud (fortaleza de la reflexión), frecuencia (numero de oscilaciones por segundo, hertz), ancho de banda (rango de frecuencias) y fase (forma relativa y posición en tiempo de una reflexión) [1].

2.1 Geófono

Los geófonos son instrumentos utilizados para convertir el movimiento de la tierra generado por un pulso sísmico en señales eléctricas. Casi todos los geófonos empleados para la prospección sísmica a partir de la superficie terrestre firme son de tipo electromagnético, constituido por una bobina y un imán. La bobina está fijada rígidamente con respecto a la tierra de tal manera, que se moverá junto con la superficie terrestre en respuesta a los movimientos sísmicos, mientras que el imán tarda en moverse.

2.2 Procesamiento sísmico

El procesamiento sísmico es un conjunto de tratamientos aplicados a los datos sísmicos adquiridos en campo con el fin de obtener secciones sísmicas de calidad, lo cual permite entender las estructuras geológicas de interés en sub-superficie [2]. El objetivo fundamental de todo procesamiento multiseñal es aislar en los registros las reflexiones de los otros eventos sísmicos que se superponen a ellas (ruido ambiental, onda aérea, etc.). Existen tres etapas en el procesamiento de datos:

- *Pre-Procesamiento.* Los datos de campo son grabados en el modo “multiplexado”. Luego estos son “demultiplexados”, es decir, se realiza la transposición de la matriz de tal modo que las columnas de la matriz resultante puedan ser leídas como las trazas sísmicas grabadas en los diferentes receptores para un punto de disparo común. Una vez obtenidos los registros de campo, con señales y ruidos, sigue la tarea de procesarlos digitalmente, mediante programas específicos. Un instrumento matemático fundamental para este fin es la Transformada de Fourier que en sísmica permite pasar del dominio del tiempo (la traza, o sea una serie de valores de amplitud a lo largo del tiempo) al dominio de la frecuencia (el espectro de frecuencia donde vemos en ordenadas las amplitudes de señal y ruido indiscriminadamente, correspondientes a cada frecuencia registrada). La transformación se realiza aproximando la forma de la traza con una integración de una serie de funciones armónicas (seno, coseno) o Serie de Fourier, de amplitudes variables, para poder entonces pasar al cálculo y representación del espectro de frecuencia (expresable también como una integración, pero ahora de diferenciales de frecuencia). Hoy en día, el proceso de demultiplexación es realizado por las empresas de adquisición sísmica. La Deconvolución, es ejecutada a lo largo del eje del tiempo, tiene como objetivo mejorar la resolución temporal, comprimiendo el pulso efectivo de la fuente contenido en la traza sísmica a un Pulso Dirac o función Delta (en literatura sísmica este pulso es conocido como un Spike, la operación de reducción del pulso sísmico a un Spike es denominada Deconvolución Impulsiva). Dado que el ruido y la señal de alta frecuencia son aumentados por este proceso, estos son frecuentemente sometidos a un filtro pasa-bandas.
- *Apilamiento.* También es un proceso de comprensión. Este proceso provee la primera imagen del subsuelo. Realiza un reordenamiento de las trazas sísmicas en conjuntos de punto reflector común o CMP (Common MidPoint). Este ordenamiento consiste en agrupar las trazas que por geometría pertenecen a un mismo punto medio entre una fuente y un receptor determinado. Se deduce, por construcción, que el espaciado entre CMP es la mitad del espaciado entre geófonos.
- *Post-apilamiento.* Etapa que usa la migración, es el proceso que mejora la resolución espacial, puede ser realizada en el dominio del tiempo o de profundidad. La migración en profundidad es conceptualmente más simple de entender, pero más difícil de ejecutar. Ella tiene por objetivo transformar los datos sísmicos registrados en tiempo en una sección en profundidad. La migración en el tiempo es más simple y menos costosa. Su objetivo es transformar los datos sísmicos registrados en tiempo en secciones, también en el dominio de tiempo, en las cuales las estructuras se parezcan más con aquellas existentes en el subsuelo. La idea es hacer que la sección en tiempo se parezca lo más posible a aquella que sería vista por un observador mirando directamente para abajo, en la dirección vertical.

Todas las técnicas de procesamiento restantes pueden ser consideradas secundarias, es decir, ayudan a mejorar la eficacia de los procesos primarios. Por ejemplo, puede ser necesaria la aplicación de un filtro para eliminar eventos según la inclinación, antes que la Deconvolución sea aplicada. Filtros pasa-bandas también pueden ser necesarios para remover el ruido de las

frecuencias bajas y altas. Antes de la Deconvolución, la corrección de la divergencia esférica es necesaria para compensar la pérdida de amplitudes debido a la propagación del frente de onda. El análisis de la velocidad, que es una etapa esencial para el apilamiento, puede ser mejorado por la supresión de las reflexiones multiplexadas y por la corrección de las estáticas de campo. La corrección de las estáticas residuales mejora también el análisis de velocidades y, de ese modo, la calidad de la sección apilada.

3 Archivos Binarios SEG-Y

El formato SEG-Y (Fig 2) ha sido adaptado por la Sociedad de Geofísicos de Exploración (SEG) como un estándar para los datos de seguimiento de una traza sísmica. En este formato los geólogos analizan el tiempo que se demora una onda en ir hasta cierta profundidad y volver a superficie, esta profundidad está directamente relacionada a la litología, es decir tipo de roca o las diferentes respuestas de varios fluidos, por ejemplo: agua, petróleo, etc. El formato está conformado por tres partes:

- Un encabezado en formato EBCDIC (Extended Binary Coded Decimal Interchange Code), que consta de 3200 byte, distribuido en 40 líneas de texto con 80 caracteres por línea, que describe las características de la cinta tales como el nombre de la zona, el nombre de la línea, parámetros de grabación, entre otros.
- Un encabezado binario de 400 bytes que contiene el número de muestras, frecuencia de muestreo, y el código de formato; además de información sobre cómo fue grabada la traza sísmica.
- Trazas sísmicas. Contiene una cabecera de 240 byte seguido de los datos.

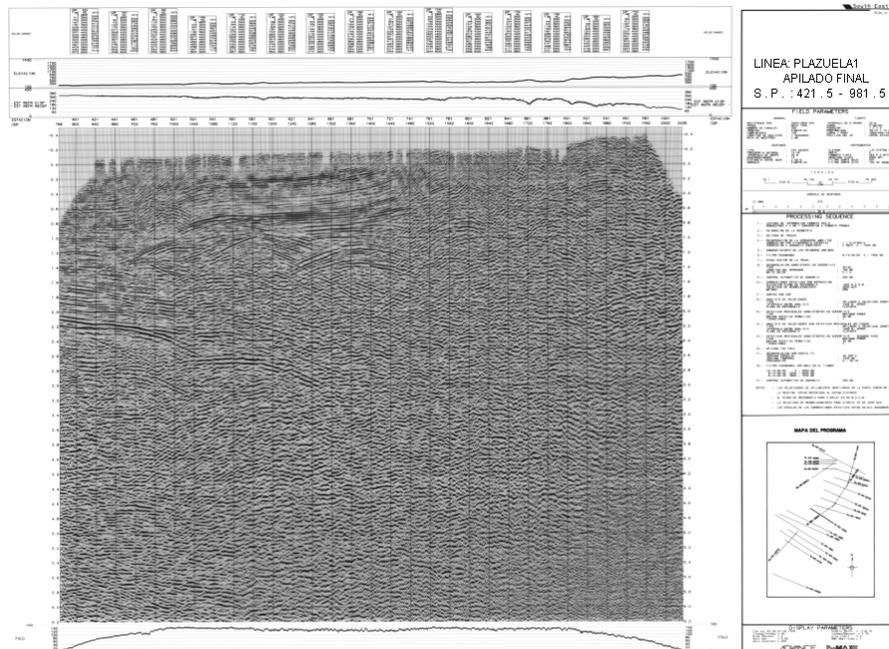


Fig. 2. Formato SEG-Y

3.1 Sísmica 2D o 3D

La sísmica 2D se utiliza para reconocimiento en áreas poco exploradas o en zonas ambientales muy sensibles, remotas o de muy difícil acceso, es más económica e impone menores restricciones logísticas y ambientales. El uso de sísmica 3D se justifica en la relación costo-beneficio en áreas donde se tenga más de un yacimiento y se requiera un modelo geológico unificado, en áreas exploratorias con exceptivas de grandes reservas, áreas de geología compleja, en casos donde la perforación de pozos exploratorios sea muy costosa y en casi todos los casos de exploración marina.

La sísmica 3D no es equivalente a la sísmica 2D, lo que realmente distingue a las sísmica 3D es el concepto de azimut (ángulo entre la fuente y receptor); en el caso 2D todas las trazas se adquieren con el mismo azimut, mientras que en la sísmica 3D para cada fuente los receptores se ubican en diferentes direcciones, lo que significa que diferentes trazas de una misma fuente tienen diferentes azimuts. El muestreo del subsuelo a partir de rayos con diferentes azimuts significa que la información sísmica que se obtiene en cada punto del subsuelo se encuentra afectada por la geología de los puntos cercanos en todas las direcciones, mientras que en la 2D no lo está.

4 VTK (Visualization Toolkit)

Visualization ToolKit (VTK) es un software de dominio público, distribuido bajo el modelo Open Source, para computación gráfica en 3D, procesamiento de imágenes y visualización [3] [4]. Los algoritmos incluyen métodos escalares, vectoriales, de tensores, de textura, volumétricos además de técnicas de modelado avanzado como la reducción de polígonos, el aplanado de mallas, recorte y triangulación de Delaunay. El diseño de esta biblioteca está fuertemente basado en objetos y las aplicaciones pueden escribirse directamente en C++, Tcl (Tool Command Language) o Java.

VTK se desarrolló a partir de dos modelos: el modelo gráfico y el modelo de visualización. El primero captura las principales características de un sistema gráfico; su objetivo es transformar datos gráficos en imágenes, es decir, los objetos que intervienen en la representación de una escena. En VTK se utilizan siete objetos básicos para renderizar una escena:

- ***vtkRenderWindow***. Gestiona una ventana en el dispositivo de la pantalla, para renderizar es decir obtener una imagen bidimensional a partir de la información de una escena tridimensional, como la geometría las luces, etc.
- ***vtkRender***. Coordina el proceso de renderizado con luces, cámaras y actores.
- ***vtkLight***. Fuente de luz para iluminar la escena.
- ***vtkCamera***. Permite definir la posición de la vista, el punto focal, y las propiedades de visualización de la escena.
- ***vtkActor***. Representa un objeto renderizado en la escena, incluyendo sus propiedades y su posición en el sistema de coordenadas mundial.
- ***vtkProperty***. Define las propiedades de apariencia de un objeto o actor como el color, la transparencia, y las propiedades de iluminación, como especular y difusa. También las propiedades de representación como de red y de superficie sólida.
- ***vtkMapper***. Representación geométrica de un actor.

El papel del modelo de visualización es transformar información en datos gráficos; esto significa que el modelo de visualización es el responsable de construir la representación geométrica que se renderiza mediante el modelo gráfico. Existen dos tipos básicos de objetos:

- **vtkDataObject.** Clase genérica que permite representar diferentes tipos de datos. Los objetos de datos consisten en estructuras geométricas y topologías (puntos y celdas), y también en atributos asociados, tales como escalares y vectoriales.
- **vtkProcessObject.** Objeto que hace referencia a filtros que actúan sobre los actores modificándolos.

En VTK, el procesamiento de los datos se realiza a través de un pipeline; el cual se puede describir como una serie de elementos unidos (figura 3) donde la salida de una transformación se enlaza con la entrada de otra. Esta arquitectura facilita el agregar o quitar transformaciones. Una vez los datos entran en el pipeline, pueden ser afectados por multitud de procesos o transformaciones. Cuando hay una petición de salida, la fuente toma los datos iniciales y los almacena, estos son enviados al filtro para ser procesados (si es necesario), luego son llevados al mapper quien se encarga de la geometría (pasar los datos a objetos), en seguida se comunica con el actor para ajustar las propiedades visibles y por último enviados a la ventana para ser ejecutados y visualizados. Para leer los datos se debe emplear el reader (lector) apropiado para el tipo de dato.

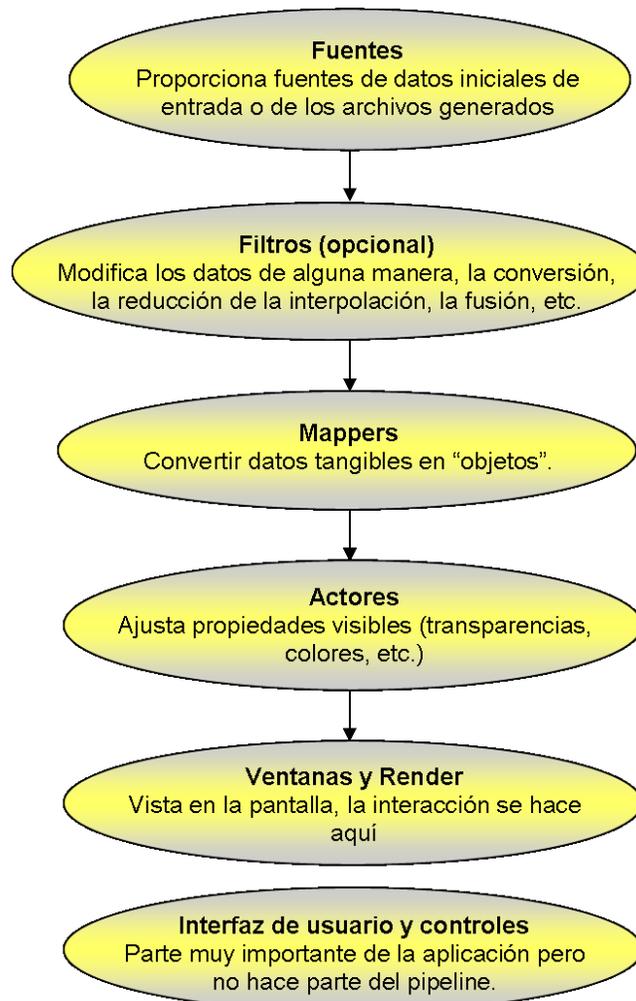


Fig 3. Diagrama Pipeline

5 Software abcSEG-Y

La herramienta *abcSEG-Y* se desarrolló a partir de las librerías de VTK (Visualization Toolkit), bajo el sistema operativo de Linux, por medio del software de desarrollo NetBeans. Para la aplicación, tanto en 2D como en 3D, se utilizaron los siguientes objetos:

- ***vtkActor***. Representa un objeto (la geometría y propiedades) en una escena renderizada.
- ***vtkCamera***. Proporciona los métodos para posicionar y orientar el punto de vista y el punto focal para una cámara virtual.
- ***vtkDataSetMapper***. Clase abstracta que especifica una interfaz para los objetos del conjunto de datos, también crea métodos para proporcionar información acerca de los datos.
- ***vtkLine***. Aplicación para representar una línea.
- ***vtkPanel***. Contenedor de objetos visuales de vtk.
- ***vtkPoints***. Aplicación para representación y manipulación de puntos en 2D y 3D.
- ***vtkRenderer***. Es el encargado de renderizar una escena, este objeto recorre las distintas fuentes de datos.
- ***vtkStructuredPoint***. Es una subclase de *vtkImageData*, esta representa una estructura geométrica, es decir una matriz regular topológica y geométrica de los puntos.
- ***vtkUnstructuredGrid***. Conjunto de datos que representa combinaciones arbitrarias de todos los tipos de celdas posibles. Las redes no estructuradas son definidas por los puntos, las celdas y los tipos de celdas.

La visualización de los datos extraídos de archivos SEG-Y 2D se realizó a partir de la función *ViewWiggle2D*, la cual genera un matriz con los puntos o datos obtenidos del archivo SEG-Y, los ordena en una grilla y guarda la información en un *vtkUnstructuredGrid*, después por medio de *vtkStructuredPoints* se crea el objeto imagen, el cual es ingresado dentro de un *vtkDataSetMapper* para finalmente ser graficado en un *vtkPanel*.

La función *ViewWiggle2D* relaciona las siguientes clases:

- ***Clase SegyTraceHead***. A partir de los encabezados se extrae la información de las trazas del archivo, la información se guarda en los parámetros del constructor leyendo cada dato dependiendo de su tipo; para esto se adaptan métodos entre los cuales se tienen: *getmt* (entero), *setmt* (entero), *toString* (string), *intBitsIBMTToFloat* (entero-float) y *floatToInBitsIBM* (float-entero). De otro lado se implementaron métodos para realizar la lectura, validación y copias de seguridad del archivo como *copyRgFileHead* que realiza una copia del archivo, *copyRgHead* realiza una copia del encabezado, *isValid - isGeogr - regularize* validan que los datos estén en un rango aceptable, esto se hace para que la imagen se genere correctamente y *write* escribe en un archivo binario la información obtenida del SEG-Y.
- ***Clase SEGYFileHead***. Se utiliza un ciclo para realizar la lectura del archivo en busca de información específica de los encabezados (trazas, cabezas, entre otras) a través del método *scanSEGYFile*, luego se obtienen datos a través de los métodos mencionados anteriormente en la clase *SegyTraceHead*. A través del método *read* se hace lectura binaria del archivo del cual se obtienen valores enteros y cortos (short).
- ***Clase RgHead***. Se obtienen parámetros específicos como direcciones base para hacer la lectura del archivo, a partir de estas direcciones y con el método *read* que realiza lectura del archivo, dependiendo del tipo de dato que corresponda.
- ***Clase RgFileHead***. Se encarga de leer el archivo binario a través de los métodos *readRgFileHead* y *read*, el primero hace una lectura global del archivo en busca de posibles errores, mientras que el segundo busca datos específicos como el número de trazas y el tiempo de obtención del dato.

- **Clase Matrix.** Cuenta con dos variables enteras correspondientes al eje X y Y (filas, columnas) y un valor doble que corresponde a la data. Se crea un método *fill*, con el cual se elimina valores negativos de la matriz, además de obtener el valor máximo existente en la matriz a partir del método *GetMax*.
- **Clase ReadData.** A partir de la ubicación del archivo, esta clase llama o invoca a las clases anteriormente descritas para realizar la lectura y obtención de datos del archivo SEG-Y a visualizar; estos atributos (trazas, cabezas y valor máximo) se guardan en un objeto de tipo Matrix en el método *fillMatrix*.
- **Clase Interfaz.** En esta clase se maneja el entorno gráfico; cada objeto cuenta con eventos y gracias a estos se invocan métodos existentes en las clases anteriormente descritas. La interfaz (Fig.4) incluye los botones *Abrir* (despliega un selector en el cual se ubica el archivo SEG-Y 2D que se desea visualizar), *Ejecutar* (invoca los métodos que visualizan el archivo SEG-Y), *Rotar* (habilita la rotación de la cámara) y *Salir*.

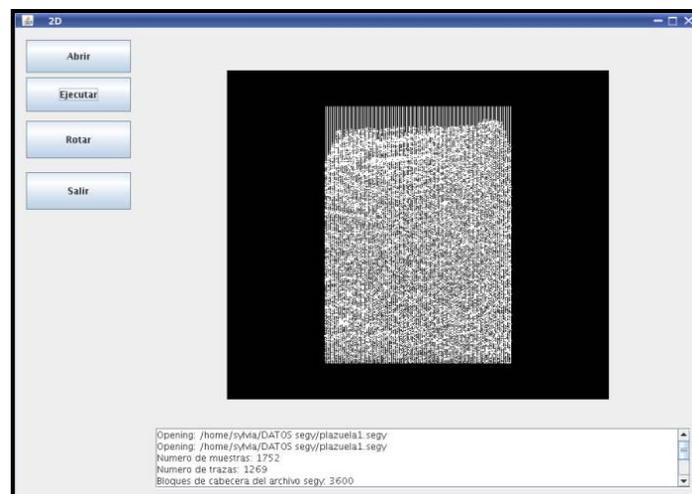


Fig. 4. Visualización archivo binario SEG-Y 2D

Para la visualización de los archivos SEG-Y en 3D se creó un método similar, *Wiggle3D*. A través de las clases de VTK se hace lectura directa del archivo y toda la información es extraída por medio de los objetos vtk, se crea un *mapper*, este es asignado a un actor al cual se le modifican propiedades como el color y opacidad, a través del *render* se cambia el fondo del actor y se activa la cámara, los datos se guardan en un objeto *vtkStructuredPoints* que realiza la lectura de los datos 3D; esta lectura se envía a un *vtkStructuredGrid* el cual organiza la información para enviarlos a un *vtkDataSetMapper* que es la estructura a ser graficada en un *vtkPanel*, encargado de visualizar el archivo. Aparte de la lectura y visualización de los datos, también son utilizados otros elementos de VTK para hacer más llamativa la gráfica.

- **MakeOutline.** Crea el contorno del gráfico 3D, es decir, los bordes del cubo.
- **MakeLookupTable.** Crea el rango de colores a partir del mapper con los cuales se realiza la visualización.
- **MakeColorBar.** Crea una barra de colores correspondiente al rango definido en el método *MakeLookupTable*, utilizado en la visualización.
- **InitCutterX.** Inicializa el plano de corte en x (crossline).
- **InitCutterY.** Inicializa el plano de corte en y (inline).
- **InitCutterZ.** Inicializa el plano de corte en z (timeslice).

- **GetCutXRang.** Obtiene el rango de valores por los cuales puede moverse el corte X.
- **GetCutYRange.** Obtiene el rango de valores por los cuales puede moverse el corte Y.
- **GetCutZRange.** Obtiene el rango de valores por los cuales puede moverse el corte Z.
- **SetPosCutX.** Cambia la posición del corte en X en una distancia que va de cero al valor obtenido por getCutXRang.
- **setPosCutY.** Cambia la posición del corte en X en una distancia que va de cero al valor obtenido por getCutYRange.
- **setPosCutZ.** Cambia la posición del corte en X en una distancia que va de cero al valor obtenido por getCutZRange.
- **makeAxes.** Crea un objeto visual ubicado en el origen de datos el cual sirve para orientar al observador dentro de la gráfica.
- **FillGrid3.** Este método crea un lector a partir del cual se lee el archivo para luego ser asignado al mapper.
- **Wiggle3D.** Utilizando los métodos anteriormente descritos, dibuja la imagen, generada a partir del SEG-Y 3D en un vtkPanel.

La interfaz incluye los botones *Abrir*, *Rotar*, *Ejecutar* y *Salir* que realizan las mismas acciones descritas anteriormente. Al visualizar el archivo se observan los planos de corte X (crossline), Y (inline), Z (timeslice) y el contorno del grafico (cubo), los cuales inicialmente están activos y se desplazan a partir de los sliders respectivos; estos ejes además se pueden ocultar.

En la figura 5 se observa un archivo SEG-Y 3D, los cambios de tonalidad representan la polaridad de la onda sísmica de acuerdo con el tipo de roca, de forma análoga como se observaría en un software comercial como Petrel (Fig.6). La herramienta presenta la limitación que no puede leer archivos con una extensión mayor a 1.000.000 de trazas, dado que el tamaño de la matriz se encuentra limitado, cuando llega a este valor se desborda.

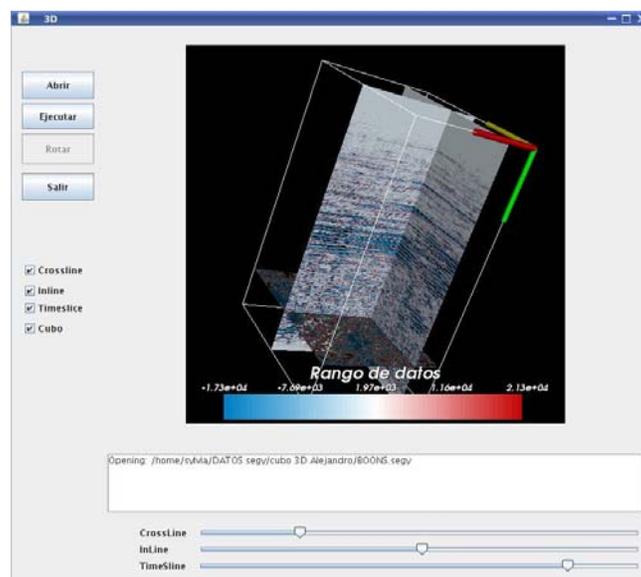


Fig. 5. Visualización archivo binario SEG-Y 3D – abcSEG-Y

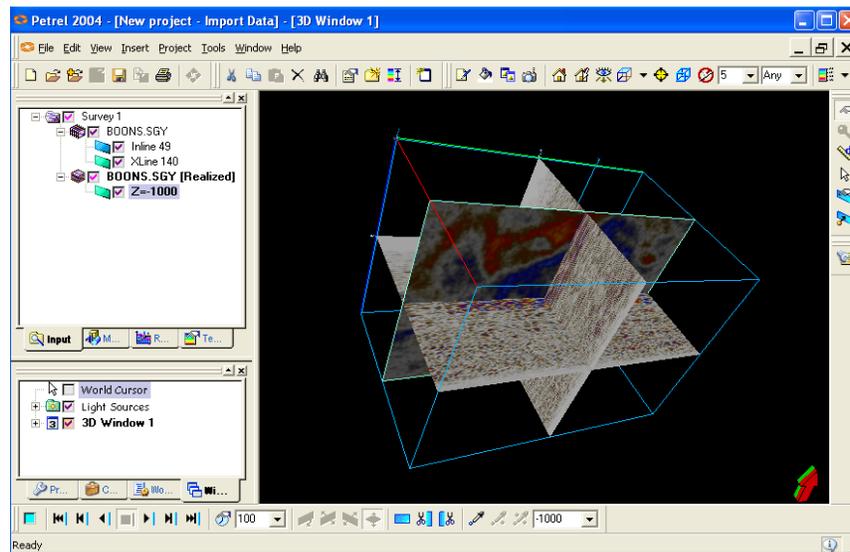


Fig. 6. Visualización archivo binario SEG-Y 3D – Petrel

Algunas compañías crean sus propios archivos para fines no comerciales sin seguir estrictamente el estándar de la SEG, es decir, crean el archivo solo con la información de las trazas sísmicas y obvian la información general como el nombre de la compañía que realiza el levantamiento, números de geófonos utilizados en la muestra, parámetros de grabación de la cinta, etc. El software *abcSEG-Y* se desarrolló con la capacidad de visualizar archivos tanto comerciales como académicos.

6 Conclusiones

El uso de herramientas de desarrollo y distribución libre es una buena alternativa a la hora de desarrollar diferentes aplicaciones que pueden ser empleadas para el desarrollo de proyectos de mayor complejidad. El software *abcSEG-Y* es una herramienta de apoyo a la investigación, capaz de visualizar tanto archivos sísmicos tanto comerciales como académicos, desarrollado a partir de las librerías de VTK.

Referencias

- [1] Alcántara, Nehomar. *Inversión de impedancias acústicas utilizando el método del gradiente espectral proyectado*. Universidad Simón Bolívar – Ingeniería Geofísica. 2006. <http://www.gc.usb.ve/geocoordweb/Tesis/Pre/Nehomar%20Alcantara.pdf>
- [2] Gamboa Fernando. *El procesamiento sísmico*. Instituto Colombiano del Petróleo – ICP - Grupo de Geofísica, Ecopetrol, 2007.
- [3] Schroeder, W.J.; Martin, K.M.; “The VTK User’s Guide”. Kitware, Inc., 1999.
- [4] Schroeder, W.J.; Martin, K.M.; Lorensen B.; “The Visualization Toolkit”. Kitware, Inc., 2006.