



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE INGENIERIA**

**CONVERSION DE INFORMACION GRAFICA  
DE SECCIONES SISMICAS EN COMANDOS  
DEL SISTEMA HSR-11B A FORMATO CGM**

**T E S I S**

**QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACION  
P R E S E N T A**

**MIGUEL ANGEL AVILA MARTINEZ**

**DIRECTOR:**

**ING. GABRIELA BETZABE LIZARRAGA RAMIREZ**



**CIUDAD UNIVERSITARIA**

**1998**

**TESIS CON  
FALLA DE OPICEM**

269231



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

---

## Definición del problema

La presente tesis describe el desarrollo de un programa de computadora encargado de procesar un conjunto de gráficas de secciones sísmicas, generadas en un computador central IBM 9121, con el propósito de efectuar su conversión a un formato gráfico vectorizado, estándar dentro de la industria e independiente de cualquier plataforma de hardware. Este trabajo forma parte de un proyecto de desarrollo de software propuesto por el Departamento de Programación y Sistemas, perteneciente a la Gerencia de Prospección Geofísica del Instituto Mexicano del Petróleo (IMP). En los siguientes párrafos se describe brevemente el propósito del mismo.

Dentro del organigrama del IMP -Institución de apoyo tecnológico a Petróleos Mexicanos (Pemex), existe una subdivisión denominada Exploración y Producción, dividida a su vez en varias gerencias. Una de ellas, la Gerencia de Prospección Geofísica, se encarga de recibir y procesar información de campo recabada por las brigadas de exploración de Pemex o de sus compañías contratistas. Dicha información consiste en una serie de datos grabados en cintas magnéticas referentes a trabajos de reconocimiento y exploración que se llevan a cabo en zonas factibles para la obtención de hidrocarburos (petróleo y gases).

Dentro de la ciencia geofísica existen diferentes técnicas de prospección que se utilizan para conocer la condición estructural del subsuelo, de entre todas ellas, los métodos de prospección sísmológica son los más comúnmente utilizados por la industria petrolera. Datos referentes a exploraciones utilizando estos métodos -en particular el denominado *método de prospección sísmológica de reflexión*, son los que se reciben y analizan en el Centro de Proceso de la gerencia antes mencionada.

En este lugar, un grupo de analistas geofísicos se encargan de procesar estos datos con ayuda de una computadora vectorial IBM 9121, la cual es utilizada, entre otras cosas, para generar una serie de gráficas en dos dimensiones con ayuda de un programa especial para el procesamiento y graficado de información sísmica denominado CAPS (Computer Aided Plotting System: Sistema de Graficación Asistido por Computadora), el cual fue desarrollado por una compañía norteamericana de exploración geofísica: la Western Geophysical Company of America. Estas gráficas (denominadas *secciones sísmicas*), son una representación bidimensional de las diversas estructuras geológicas en profundidad, es decir, lo que se obtiene es una especie de "radiografía" de la ubicación de las diferentes capas que se encuentran por debajo de la superficie terrestre en una dirección determinada denominada *línea de tiro*.

Las gráficas así obtenidas, para diferentes líneas de tiro y dispuestas en un sistema de cuadrícula, se utilizan posteriormente para construir con ayuda de herramientas automatizadas un modelo tridimensional de la condición estructural del subsuelo -proceso conocido como análisis en 3D, como último paso previo a su interpretación geológica para la deducción de aquellas zonas en las que pudiera encontrarse petróleo o gas.

El programa desarrollado en la presente tesis tiene como objetivo efectuar un proceso sobre las gráficas generadas en el computador central -las cuales hasta fecha reciente únicamente se imprimían en papel o en película semitransparente, para convertirlas a un estándar gráfico vectorizado denominado Metaarchivo para Gráficos por Computadora (CGM: Computer Graphics Metafile).

Una vez convertidas y almacenadas como archivos en formato CGM, estas imágenes podrán ser utilizadas por otras aplicaciones como un auxiliar en la generación de las secciones sísmicas en 3D. Estos programas no necesariamente tendrán que ser específicos para la computadora principal, ya que la Gerencia de Prospección Geofísica cuenta además con un sistema paralelo IBM SP2 9076, al cual se encuentran conectadas varias terminales gráficas que hacen uso de dos programas de procesamiento de información sísmica: OMEGA, de la Compañía Western Geophysical y PROMAX de la Landmark Advance Geophysical Corporation. Ambos paquetes tienen la capacidad de leer archivos gráficos en formato CGM, con lo cual el sistema de conversión servirá como un software de interfaz entre los procesos hechos en la *mainframe* IBM 9121 y aquellos que se lleven a cabo en el sistema SP2.

Por otra parte, en varias ciudades ubicadas en las principales zonas petroleras del país (Villahermosa, Tabasco; Cd. del Carmen, Campeche; Poza Rica, Veracruz), la división de Pemex-Exploración cuenta con Centros de Proceso en los que dispone de equipo similar al utilizado en la Gerencia de Prospección Geofísica del IMP.

Anteriormente el resultado de un proceso de información sísmica hecho dentro del Instituto, se enviaba a los Centro de Proceso de dichas ciudades en forma impresa (las gráficas generadas por CAPS). Ahora, con el sistema propuesto, se tendrá la ventaja de poder enviar estos resultados en forma de archivos gráficos en formato CGM almacenados en cintas magnéticas, lo que elimina la necesidad del envío de copias impresas. De esta manera cada Centro de Proceso podrá imprimir las gráficas que requiera utilizando sus propios equipos de cómputo

---

## Reconocimientos

Varias personas han contribuido con su invaluable ayuda en la realización de este trabajo. Quisiera agradecer en especial el apoyo brindado por el siguiente personal del Instituto Mexicano del Petróleo: M. C. Héctor Palafox Rayón, titular de la Gerencia de Prospección Geofísica; Ing. José Jaime Ruíz Hernández, jefe de división de la Línea de Adquisición y Procesamiento de Datos Geofísicos; Ing. Evererado Matus Puga, titular del Departamento de Programación y Sistemas; Ing. Arturo Díaz Bustos, quien fungió como asesor interno dentro del Instituto; el personal de operación del Centro de Proceso y el grupo de analistas geofísicos pertenecientes a la misma división. Su asesoramiento, corrección y puntos de vista, en especial en aquellas partes del trabajo que están fuera del ámbito de mi especialidad, me sirvieron de guía para entender mejor el contexto en el cual opera el programa desarrollado en la presente tesis. Asimismo, a mis familiares, profesores de la Facultad y a la Universidad mi más sincero agradecimiento por haberme dado la oportunidad de cursar estudios a nivel superior, a todos ellos dedico este trabajo.

Miguel A. Avila Martínez.

---

# Indice de contenido

Tema	Página
<b>Parte I. Conceptos preliminares sobre exploración de hidrocarburos.</b>	
<b>Capítulo 1 La ciencia geofísica y sus métodos de prospección.....</b>	<b>1</b>
1.1 Métodos por gravedad .....	3
1.2 Métodos magnéticos .....	3
1.3 Métodos eléctricos .....	4
1.4 Método de prospección por radiactividad.....	4
1.5 Métodos de prospección sísmica .....	5
1.6 Comparación de los métodos más importantes de prospección geofísica .....	6
<b>Capítulo 2 El método de prospección sísmológica de reflexión .....</b>	<b>8</b>
2.1 Instrumentos utilizados en los trabajos de prospección sísmica.....	9
2.2 Operaciones sísmicas en tierra.....	13
2.3 Operaciones en zonas cubiertas por el agua .....	16
2.4 Geometría de las trayectorias de reflexión en superficies de separación horizontales e inclinadas.....	17
2.5 Diagramas y perfiles de los datos de reflexión.....	19
<b>Parte II. Antecedentes en el desarrollo del sistema.</b>	
<b>Capítulo 3 Secuencia de proceso de la información de campo en la Gerencia de Prospección Geofísica del Instituto Mexicano del Petróleo .....</b>	<b>20</b>
3.1 Estructura orgánica del Instituto Mexicano del Petróleo.....	21
3.1.1 Subdirección de Exploración y Producción .....	21
3.1.2 Gerencia de Prospección Geofísica .....	25
3.2 El Centro de Procesamiento Geofísico .....	28

Tema	Página
Capítulo 4 Descripción del sistema rasterizador HSR-11B .....	30
4.1 Formato general de los comandos .....	32
4.2 Definición de los comandos .....	33
4.2.1 No operation .....	33
4.2.2 Seismic traces .....	34
4.2.3 Timming line .....	40
4.2.4 Serial data .....	41
4.2.5 Seismic flush .....	42
4.2.6 Graphic flush .....	43
4.2.7 Restore raster buffer .....	44
4.2.8 Sectored vectors.....	45
4.2.9 Scan line advance .....	47
4.2.10 Initialize .....	48
4.2.11 Alphanumeric.....	49
4.2.12 Resampling .....	52
Capítulo 5 Formato gráfico CGM (Computer Graphics Metafile).....	54
5.1 Estructura del metaarchivo .....	55
5.2 Clases de elementos .....	57
5.2.1 Clase 0: elementos delimitadores .....	57
5.2.2 Clase 1: descripción de elementos del metaarchivo .....	59
5.2.3 Clase 2: descripción de elementos gráficos.....	61
5.2.4 Clase 3: elementos de control.....	63
5.2.5 Clase 4: elementos primitivos de la gráfica.....	64
5.2.6 Clase 5: atributos de los elementos .....	65
5.2.7 Clase 6: elementos de escape .....	68
5.2.8 Clase 7: elementos externos .....	68
5.3 Formas de codificación.....	68
5.3.1 Metaarchivo binario.....	69
Tipos y tamaños de datos .....	70
Ejemplos de codificación .....	75
5.3.2 Metaarchivo en modo de texto .....	77
Abreviaturas utilizadas .....	77
Ejemplo de codificación .....	80
5.3.3 Metaarchivo en modo carácter .....	86
Códigos de operación .....	86

Tema	Página
<b>Parte III. Desarrollo del sistema para la creación de archivos gráficos en formato CGM a partir de información sísmica.</b>	
<b>Capítulo 6 Fase de definición .....</b>	<b>88</b>
6.1 Análisis del sistema .....	93
6.1.1 Asignaciones.....	98
6.2 Esquema del plan del proyecto de software .....	101
6.2.1 Alcance.....	101
6.2.2 Recursos .....	102
6.2.3 Costos .....	103
6.2.4 Plan temporal.....	103
6.3 Análisis de requerimientos .....	104
6.3.1 Descripción de la información.....	104
6.3.2 Criterios de validación.....	108
6.3.3 Manual de usuario preliminar.....	108
 <b>Capítulo 7 Fase de desarrollo .....</b>	 <b>112</b>
7.1 Diseño del software .....	113
7.1.1 Diseño arquitectónico y de datos.....	114
7.1.2 Diseño procedimental .....	127
7.2 Codificación.....	138
7.3 Implantación y prueba final del sistema .....	156
7.3.1 Ejemplos de las gráficas de secciones sísmicas generadas por el sistema de conversión.....	158
 <b>Conclusiones .....</b>	 <b>159</b>
<b>Glosario de términos .....</b>	<b>161</b>
<b>Bibliografía.....</b>	<b>164</b>



---

# Introducción

La prospección geofísica es el procedimiento de búsqueda de yacimientos minerales o depósitos ocultos de hidrocarburos (petróleo y gases) efectuando mediciones físicas desde la superficie del suelo, mediciones que, de ordinario, suministran información acerca de las diversas propiedades de los materiales del interior de la Tierra.

Esta rama de la geofísica ofrece diferentes métodos de prospección del subsuelo, entre los que destacan por su importancia los métodos por gravedad, magnéticos, eléctricos, los métodos de prospección por radiactividad y los métodos de refracción y reflexión sísmica, siendo este último el de mayor uso en la exploración del petróleo, ya que proporciona una mejor información estructural del interior de la Tierra que cualquiera de los restantes procedimientos.

Los hidrocarburos son de baja densidad, por lo que dentro de las capas de la Tierra tienden a subir, así que para que exista un depósito se requiere la presencia de una capa o roca sello que impida que los hidrocarburos se escapen hasta la superficie de la Tierra. Inmediatamente debajo de esta roca sello, debe haber una roca porosa donde los hidrocarburos puedan acumularse.

Cabe aclarar que el petróleo en el subsuelo no se encuentra en "charcos", "lagos" o "cavernas", sino que está dentro de los poros de las rocas, semejante a un líquido retenido en una esponja o como el agua absorbida por un ladrillo de la construcción. Con los métodos geofísicos de prospección por gravedad y magnéticos se obtienen planos regionales que nos indican la posición y la profundidad de las rocas de basamento, es decir, aquellas que soportan a las que contienen el petróleo. Sin embargo, son los métodos de prospección sísmológica los que proporcionan en detalle una mayor información sobre las capas que pueden contener hidrocarburos, de ahí su importancia y amplio uso.

Estos métodos, en lo que se refiere a los procedimientos de campo empleados en la exploración, consisten en la generación de una energía mecánica en forma de ondas estáticas cercanas a la superficie de la Tierra, energía que viaja en todas direcciones a través de las capas de la corteza terrestre. Debido a las diferentes condiciones físicas de esas capas, parte de la energía se refleja en cada una de ellas hacia la superficie, donde previamente se han colocado unos detectores de sismicidad que captan la señal reflejada. La generación de energía se hace por medio de vibradores o pequeñas explosiones de dinamita en trabajos terrestres, o bien, para el caso de prospecciones marinas, la energía se genera liberando aire o gas a altas presiones. Los registros producidos por estas señales se graban en medios magnéticos para su posterior análisis e interpretación (ver Figura 1).

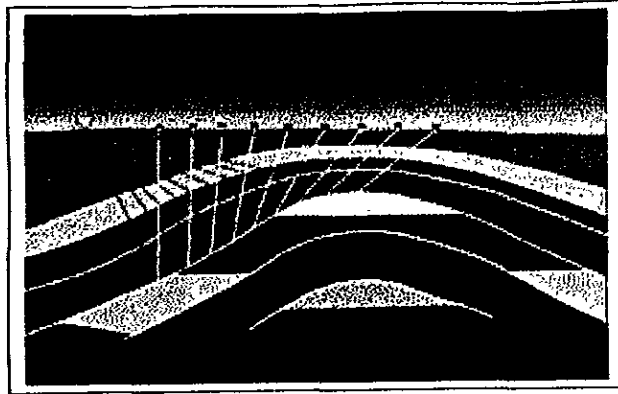


Figura 1. En los trabajos de exploración de petróleo y gas mediante los métodos de prospección sísmica, una fuente de energía acústica es disparada a nivel del suelo hacia el interior de la Tierra. El reflejo de dicha energía desde las capas en profundidad es detectado por un conjunto de sismógrafos colocados en la superficie.

Cada generación de energía produce una serie de registros sísmicos en forma de ondas que parecen encajar unas en otras, y que corresponden a las diferentes capas del interior de la Tierra.

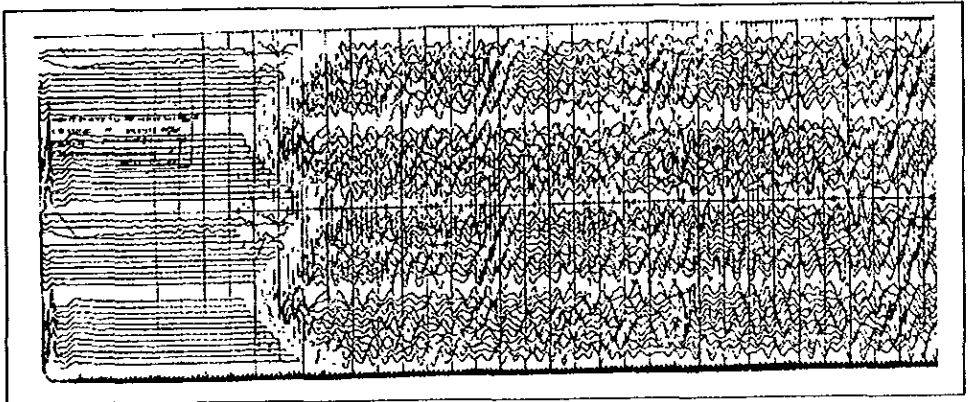


Figura 2. Registro típico de reflexión sísmica.

Si tenemos una sucesión de registros en forma secuencial (por ejemplo cada 80 o 100 metros) por varios kilómetros (de 10 a 100), se puede conocer la posición de las capas del subsuelo en direcciones predeterminadas. De esta manera, al disponer de líneas observadas en el campo en un sistema de cuadrícula, es posible inferir la posición de esas capas en un área geográfica grande.

Lo que de hecho se obtiene es una gráfica muy similar a un mapa topográfico, en el cual para una capa determinada en el subsuelo se puede observar donde hay "cerros", "hondonadas" o "fallas", es decir, conocer su condición estructural. Con este plano regional, como ya se mencionó, es posible deducir aquellos lugares favorables para la acumulación del petróleo. Hasta la fecha no existe un método que en forma directa indique con precisión la existencia de estos recursos en el subsuelo, el explorador debe deducir de la información geológica los lugares en donde el petróleo o gas se puedan acumular.

Dentro del territorio nacional, estos estudios son de fundamental importancia para la industria petrolera, ya que a partir de ellos es posible localizar zonas o regiones prometedoras para la obtención de hidrocarburos. El trabajo de adquisición de datos de campo lo realiza Petróleos Mexicanos (Pemex) por medio de sus brigadas o compañías contratistas.

Posteriormente, en centros de investigación como lo es la Gerencia de Prospección Geofísica del Instituto Mexicano del Petróleo, se recibe la anterior información de campo y se efectúa su análisis en su Centro de Proceso. Este estudio es llevado a cabo por un grupo de analistas geofísicos y consiste básicamente en:

- a) Efectuar una serie de correcciones para poner los datos a un mismo nivel de referencia.
- b) Estudiar las velocidades de transmisión de las ondas para corregir la información y determinar su profundidad.
- c) Aplicar filtros para limpiar los ruidos y señales indeseables.
- d) Aplicar procesos para mejorar los resultados y obtener *secciones sísmicas* -gráficas que visualizan la información sísmica en el plano, y que muestran con la mayor claridad posible la posición de las diferentes capas del subsuelo.

Finalmente, al término del procesamiento de la información de campo, y con ayuda de herramientas automatizadas, se genera una gráfica de la sección sísmica correspondiente.

Esta gráfica es la representación en dos dimensiones (2D) de un "corte" vertical de la corteza terrestre. Su construcción se hace a partir de la información de campo recabada por las brigadas de exploración, junto con diferentes programas para el procesamiento y graficado de información sísmica, diseñados especialmente para combinar estos datos con imágenes creadas por los analistas (por ejemplo sellos descriptivos, tablas de velocidades, de coordenadas, etc.), que proporcionan una descripción más detallada de la región en donde se hizo el estudio de prospección. En la siguiente figura se observa una gráfica de este tipo.

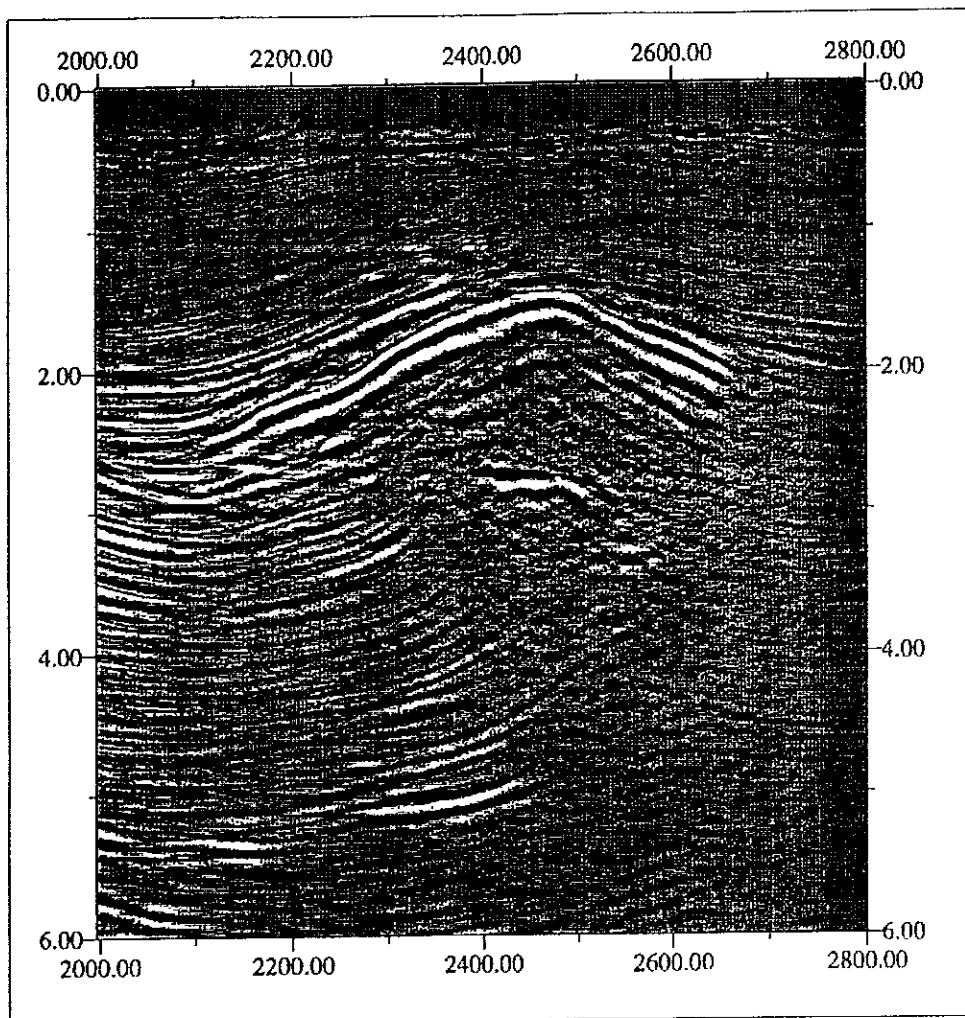


Figura 1. Gráfica de sección sísmica en dos dimensiones.

En el caso de la Gerencia de Prospección Geofísica, este proceso se efectúa en una computadora vectorial IBM 9121 con ayuda de un programa especial para el procesamiento y graficado de información sísmica denominado CAPS (Computer Aided Plotting System.- Sistema de Graficación Asistido por Computadora).

A continuación, al analizar varias gráficas de secciones sísmicas, todas correspondientes a una misma zona, área y prospecto de estudio, pero de diferentes líneas y dispuestas en un sistema de cuadrícula, es posible inferir la orientación de las capas dentro de la Tierra en un volumen preestablecido. Esto se conoce como análisis en tres dimensiones (3D) y para ello se emplean, además de las gráficas anteriores en 2D, diversas técnicas matemáticas de extrapolación para el cálculo de los puntos intermedios entre dos trazas consecutivas. La siguiente figura muestra un ejemplo de las gráficas que se obtienen después de este proceso.

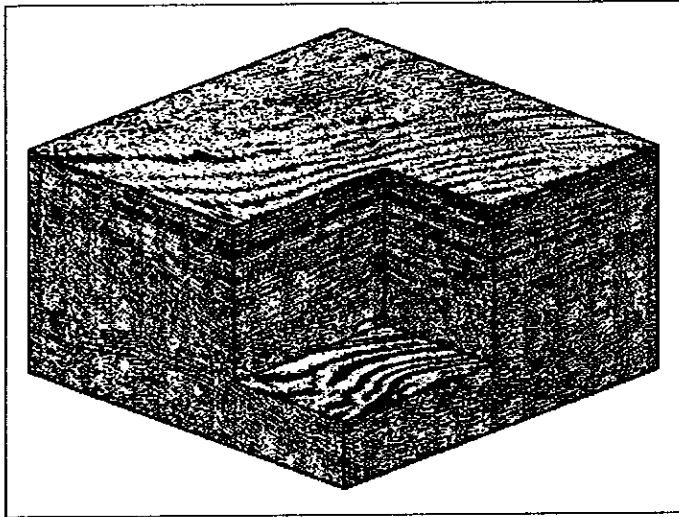


Figura 2. Corte esquemático de un modelo tridimensional del subsuelo obtenido a partir de información sísmica en dos dimensiones.

Una imagen como la anterior permite ver con mayor detalle la condición estructural del subsuelo. Las regiones continuas de color claro y oscuro representan capas dentro de la corteza terrestre. Normalmente se emplean diversos colores para diferenciarlas, de esta manera tanto el geólogo como el geofísico pueden tener una mejor referencia para ubicar aquellos lugares favorables para la acumulación de reservas de hidrocarburos.

La construcción de estas secciones es un proceso largo y complejo, el cual requiere mucha pericia por parte del analista al construir el "cubo" en 3D y hasta los años setenta este trabajo era relativamente lento y laborioso, al no disponerse de herramientas automáticas para llevarlo a cabo.

---

Sin embargo, debido al rápido avance tecnológico que a partir de la pasada década se ha experimentado en el campo de las computadoras -al disponerse hoy en día de procesadores más veloces y con una mayor capacidad de ejecución, la tarea del procesamiento de la información sísmica se ha beneficiado enormemente, ya que actualmente se dispone de equipos y de programas de cómputo que permiten reconstruir más rápido y en forma automática las secciones sísmicas tridimensionales a partir de la información en 2D, lo que anteriormente no se podía llevar a cabo en un tiempo razonable dado el cálculo numérico intensivo que se requería para ello.

El presente trabajo de tesis, describe el desarrollo de un programa de computadora encargado de procesar las imágenes bidimensionales de las secciones sísmicas generadas a partir de CAPS. El objetivo del sistema es convertir dichas imágenes a un formato gráfico estándar dentro de la industria petrolera e independiente de cualquier plataforma de hardware, denominado Metaarchivo para Gráficos por Computadora (CGM.- Computer Graphics Metafile), el cual está concebido especialmente para el almacenamiento y recuperación de información gráfica vectorizada.

El propósito del sistema es el de servir como un software de interfaz que permita compartir la información gráfica generada en el computador central IBM 9121, para utilizarla como auxiliar en la construcción de las secciones sísmicas en 3D, con ayuda de otros programas de aplicación en otras plataformas de cómputo, como ya se mencionó en la parte correspondiente a la Definición del Problema. Adicionalmente, este sistema servirá también para agilizar el envío de resultados en forma gráfica de los procesos sísmicos hechos dentro de la Gerencia, hacia otros centros de cómputo pertenecientes a la división de Pemex-Exploración ubicados en diferentes zonas petroleras del país.

---

## **Parte I**

# **Conceptos preliminares sobre exploración de hidrocarburos**

---

# Capítulo 1

## Introducción a la ciencia geofísica y sus métodos de prospección

La geofísica es la ciencia que aplica los métodos de la física al estudio de la Tierra, desde su interior hasta las altas regiones de la atmósfera; abarca un campo muy vasto, incluyendo el estudio de las ondas sísmicas, la formación de las montañas, la vulcanología, temperatura, campo magnético, propiedades de los diversos estratos, etc. Esta ciencia como tal, esta incluida dentro de un grupo mayor al que se le conoce como *geociencias* (ver figura 1.1), y que se forman a partir de la relación entre la geología y otras ciencias como la química (geoquímica), las matemáticas (geoestadística) y la física (geofísica), y que en su conjunto estudian nuestro planeta desde diferentes puntos de vista.

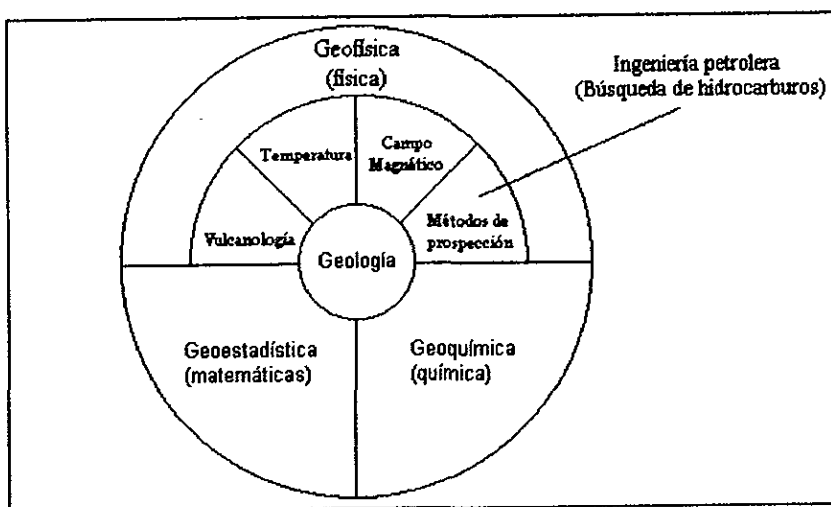


Figura 1.1. Geociencias. La búsqueda de recursos energéticos del subsuelo se lleva a cabo con el auxilio de los métodos de prospección geofísica.

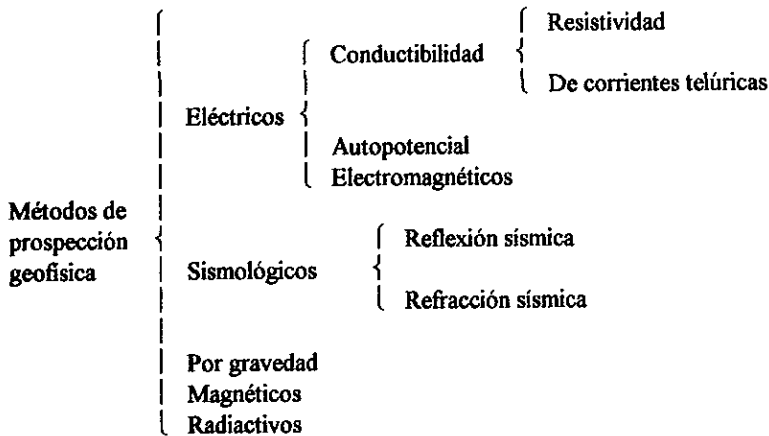
La geología ha estado siempre limitada a las observaciones recogidas en la superficie del suelo o en unos pocos sondeos, pero modernamente ha desarrollado otros medios de conocimiento. El auge de las nuevas técnicas, extraordinariamente rápido, se sitúa a comienzos del presente siglo y dio origen a una nueva ciencia: la geofísica propiamente dicha.



A diferencia del geólogo, el geofísico nunca ve los objetos que estudia; sólo mide algunas de sus características y por ellas los identifica. El geólogo diferencia dos muestras de roca por el color, por la textura, por la naturaleza (arenisca, caliza, granito, etc.) o por los fósiles que eventualmente contenga cada una de ellas. El geofísico, en cambio, sólo mide la densidad, la resistencia eléctrica, la radiactividad, el magnetismo, la velocidad a que se propaga en ella una onda de choque, etc. Y con estas medidas consigue diferenciar dos rocas que nunca ha visto. Por eso es tan importante su especialidad a la hora de estudiar rocas profundas, inaccesibles al geólogo.

La ciencia geofísica ha tenido durante las últimas décadas una importancia relevante, debido principalmente a la creciente demanda de recursos energéticos tales como hidrocarburos o depósitos minerales de valor económico, ya que previo a su explotación, antes deben ser localizados aquellos lugares favorables para su acumulación. Es por ello que esta ciencia ofrece diferentes técnicas de búsqueda, clasificadas genéricamente como métodos de prospección geofísica y que constituyen valiosos auxiliares en la identificación de dichas zonas. Estos métodos se apoyan en un conjunto de mediciones hechas desde la superficie del suelo, mediciones que, de ordinario, suministran información acerca de las propiedades físicas de los materiales del interior de la tierra. Posteriormente esta información, interpretada de forma adecuada, puede utilizarse para localizar dichos recursos con un alto grado de confiabilidad.

Todos los métodos geofísicos de prospección están encaminados a localizar estructuras geológicas favorables para depósitos de valor comercial; pero estos métodos pueden no ser capaces de encontrar directamente los depósitos. Por ejemplo, para el caso particular de la prospección petrolífera, cuando el petróleo se encuentra retenido en trampas estructurales, estos métodos pueden resultar satisfactorios, pero si se trata de acumulaciones estratigráficas, son a veces difíciles, y con frecuencia imposibles de localizar geofísicamente. El siguiente cuadro sinóptico muestra las diferentes técnicas geofísicas de exploración del subsuelo.



A continuación se explican brevemente los diferentes métodos geofísicos de prospección, haciendo especial énfasis en los métodos de prospección sísmica para la búsqueda de hidrocarburos, ya que el sistema a desarrollar se encargará de procesar información gráfica de este tipo. Al final del capítulo se incluye un cuadro comparativo de todos los métodos de prospección, junto con las aplicaciones hacia las cuales van encaminados cada uno de ellos.

## 1.1 Métodos por gravedad.

En la prospección por gravedad se miden las pequeñísimas variaciones que en la atracción gravitatoria ejercen las rocas emplazadas en los primeros kilómetros por debajo de la superficie del suelo. Los diferentes tipos de rocas tienen densidades diferentes y las rocas más densas ejercen mayor atracción gravitacional. Si las rocas más densas están arqueadas hacia arriba, formando una *elevación estructural*, tal como un *anticlinal*, el campo gravitatorio terrestre será mayor sobre el eje de la estructura que a lo largo de sus flancos. Por otra parte, un *domo salino* que es menos denso que las rocas en que está intruido puede ser descubierto gracias a los bajos valores de la gravedad que normalmente son registrados sobre el mismo. Las anomalías de la gravedad buscadas en la exploración petrolífera pueden representar tan solo una millonésima, y hasta una diezmillonésima, del campo total terrestre. Por esta razón, los instrumentos empleados han de ser extremadamente sensibles, y los gravímetros modernos permiten descubrir variaciones de la gravedad hasta de una cienmillonésima del campo terrestre.

## 1.2 Métodos magnéticos.

La prospección magnética determina las variaciones del campo magnético terrestre atribuibles a cambios de estructura, o de la *susceptibilidad magnética* de algunas rocas próximas a la superficie. Las *rocas sedimentarias* presentan, en general, una susceptibilidad muy pequeña en comparación con las *ígneas* o *metamórficas*, y la mayoría de las exploraciones magnéticas están encaminadas a levantar el mapa de la estructura sobre o dentro del *basamento* o a descubrir directamente minerales magnéticos.

El método magnético resulta útil para la búsqueda de petróleo cuando las estructuras de las capas sedimentarias petrolíferas están regidas por características topográficas tales como crestas o fallas sobre la superficie del basamento. Las anomalías magnéticas a partir de la parte superior del basamento pueden aportar información relativa a la estructura de las capas superiores. Sucede con frecuencia que resulta difícil distinguir las anomalías magnéticas debidas a la topografía del basamento de las que son consecuencia de cambios laterales en la composición de la roca de basamento, y esta ambigüedad limita la eficacia del método. La mayor parte de la prospección magnética se realiza en la actualidad con instrumentos montados en aviones.

### 1.3 Métodos eléctricos.

Existen varias técnicas geofísicas destinadas a detectar anomalías en las propiedades eléctricas de las rocas, tales como la conductibilidad, autopotencial y respuesta a la inducción. A base de estas anomalías puede resultar posible localizar minerales que ofrezcan características eléctricas distintivas, o en su caso levantar el mapa de características estructurales asociadas a yacimientos de petróleo o de minerales. El método de resistividad se emplea para determinar variaciones laterales y verticales de la conductibilidad en el interior del suelo, y se utiliza con frecuencia para medir la profundidad a que se encuentra la roca firme en conexión con proyectos de ingeniería civil, dado que, normalmente, existe un gran contraste entre la conductibilidad de la roca firme y los materiales no consolidados que la cubren. El método de corrientes telúricas aprovecha como fuente las corrientes terrestres naturales en lugar de corrientes engendradas artificialmente e introducidas en el suelo.

El método autopotencial se utiliza para detectar las presencias de ciertos minerales que reaccionan con electrólitos en el suelo, engendrando potenciales electroquímicos. Una masa de sulfuros que aparezca más oxidada a poca profundidad que a gran profundidad engendrará potenciales de este tipo que pueden ser registrados por electrodos situados en la superficie.

Los métodos electromagnéticos detectan anomalías en las propiedades inductoras de las rocas del subsuelo. Se introduce en el suelo una corriente alterna, por lo general de alta frecuencia, y sobre la superficie o en el aire se miden la intensidad y el desfase de los potenciales inducidos por las rocas enterradas. Muchas *menas* de metales comunes engendran corrientes inducidas de intensidad muy superior a las de las rocas circundantes. La mayor parte de la prospección electromagnética se realiza hoy en día desde el aire.

### 1.4 Método de prospección por radiactividad.

La actual necesidad de encontrar materias primas fisiónables para ser usadas en los reactores nucleares ha determinado que este método de prospección (en especial para la localización de depósitos de uranio), haya tenido un auge único en la historia de la exploración minera. La mayor parte de esta actividad ha involucrado el empleo de instrumentos geofísicos, es decir, de detectores de radiaciones como los *contadores Geiger* o los *escintilómetros*. El bajo coste de algunos de estos aparatos ha dado como resultado una labor geofísica la cual ha tenido éxito satisfactorio. Gran parte de la exploración del uranio ha sido realizada desde aviones, con empleo de escintilómetros especialmente adaptados para este uso. De todos los métodos geofísicos, los de radiactividad son los que tienen la menor penetración, puesto que dichas radiaciones son absorbidas por menos de noventa centímetros de tierra que cubre el material radiactivo.

## 1.5 Métodos de prospección sísmica.

Estos métodos abarcan dos técnicas semejantes en lo que se refiere a los procedimientos de campo empleados en la exploración, pero algo distintos en cuanto a los resultados que se desea obtener. La primera de ellas es conocida como *prospección sismológica de reflexión*, y su objetivo es el de levantar el mapa de la estructura del subsuelo, haciendo uso de los tiempos requeridos por una onda sísmica engendrada por una explosión de dinamita en el subsuelo, próxima a la superficie, para volver a esta después de ser reflejada en las formaciones mismas.

Las reflexiones son registradas por instrumentos detectores colocados sobre la superficie, cerca del punto de explosión y que responden al movimiento del suelo. Dichos instrumentos reciben el nombre de *geófonos* para el caso de prospecciones terrestres, o bien *hidrófonos* para las prospecciones marinas. Las variaciones en los tiempos de reflexión de un lugar a otro de la superficie indican, por lo general, características estructurales de las rocas del subsuelo. Las profundidades hasta las superficies reflectoras pueden ser determinadas a base de los tiempos, si es posible medir la rapidez de las ondas sísmicas en la zona situada encima de aquellas. Normalmente, y con una sola explosión, pueden ser observadas reflexiones hasta profundidades de 6000 metros, de manera que en la mayoría de las zonas puede determinarse la estructura geológica de toda la sección sedimentaria. Esta técnica proporciona más información estructural y mejor que cualquier otro método geofísico, pero presenta la desventaja de que es más lenta y costosa que la mayoría de los restantes métodos. Además, son muchas las regiones en donde las reflexiones sólo pueden obtenerse con grandes dificultades.

La segunda técnica dentro de los métodos de prospección sísmica se conoce como *prospección sismológica de refracción*, aquí los instrumentos detectores se disponen a cierta distancia del punto de explosión, que es larga en comparación con la profundidad a que se encuentre el *horizonte* que haya de ser marcado en el mapa. Las ondas explosivas recorren grandes distancias horizontales a través del suelo, y el tiempo requerido para su desplazamiento informa acerca de la velocidad y profundidad de ciertas formaciones del subsuelo. Aunque el método de refracción no da tanta información, ni tan precisa, del cuadro estructural como el de reflexión, proporciona datos de la velocidad en las capas refractantes que, con frecuencia, permiten al geólogo identificarlas o especificar su *litología*. Por lo general, este método hace posible cubrir una zona dada en menos tiempo que con el método de reflexión.

En el próximo capítulo se profundiza más en el primero de los métodos sismológicos: el de reflexión, ya que a partir del mismo se obtendrá la información gráfica que será procesada por el software desarrollado en la presente tesis.

### 1.6 Comparación de los métodos más importantes de prospección geofísica.

	Refracción sísmica	Reflexión sísmica	Gravedad
Principales aplicaciones	Reconocimiento exploratorio del petróleo. Geología ingenieril. Estudios de geología regional.	Exploración detallada del petróleo	Reconocimiento exploratorio del petróleo. Estudios de geología regional.
Principales instrumentos de uso corriente	Geófonos. Sistemas de registro.	Geófonos. Sistemas de registro.	Gravímetro.
Cantidades realmente medidas	Tiempo requerido por la onda explosiva para volver a la superficie después de ser refractada por las formaciones del subsuelo.	Tiempo requerido por la onda explosiva para volver a la superficie después de ser reflejada por las formaciones del subsuelo.	Variaciones del campo gravitatorio terrestre atribuibles a estructuras geológicas.
Cantidades calculadas con las medidas	Profundidades hasta los horizontes refractantes; velocidades horizontales de las ondas sísmicas.	Profundidades hasta los horizontes refractantes; buzamientos.	Contrastes de densidades de las rocas; profundidades hasta zonas de densidad anómala.
Rasgos geológicos o económicos buscados con el método	Anticlinales, fallas, domos salinos.	Trampas estructurales del petróleo de todas clases, arrecifes.	Domos salinos, ejes o anticlinales, crestas enterradas.
Correcciones aplicadas a los datos	Meteorización, altura, intervalo "inclinación-valle".	Meteorización, altura, corrimiento de filtros.	Latitud, aire libre, Bouger, terreno.
Equipo (número de hombres)	15 o más	11-20	5
¿Se pueden hacer mediciones desde aviones?	No	No	No
¿Se emplea el método en aguas litorales?	Sí	Sí	Sí

	Magnéticos	Eléctricos	Radiactividad
Principales aplicaciones	Exploración minera. Reconocimiento exploratorio del petróleo.	Exploración minera. Geología ingenieril.	Exploración de minerales radiactivos
Principales instrumentos de uso corriente	Balanza magnética, magnetómetro discriminador de flujo, magnetómetro de resonancia nuclear.	Electrodos, potenciómetros, bobinas detectoras.	Contador Geiger, escintilómetro.
Cantidades realmente medidas	Variaciones de los elementos magnéticos atribuibles a estructuras geológicas.	Potenciales naturales, corriente transmitida entre electrodos como consecuencia de la caída de potencial.	Radiactividad natural de los materiales del suelo.
Cantidades calculadas con las medidas	Contrastes de susceptibilidad de rocas; profundidades aproximadas hasta zonas de magnetización anómala.	Resistividades de capas; profundidades aproximadas de las superficies de contacto entre capas con contraste de resistividad.	Contenido en uranio de las rocas.
Rasgos geológicos o económicos buscados con el método	Topografía del basamento, depósitos de menas magnéticas, diques y rasgos ígneos similares.	Depósitos de menas que tengan propiedades eléctricas anómalas, profundidad hasta la roca firme, profundidad hasta la capa freática.	Yacimientos de uranio.
Correcciones aplicadas a los datos	Variación diurna normal.		
Equipo (número de hombres)	3 (en tierra)	2 ó 3	1-4
¿Se pueden hacer mediciones desde aviones?	Sí	Sí	Sí
¿Se emplea el método en aguas litorales?	Sí	Sí	No

La anterior tabla fue tomada de la obra de Milton B. Dobrin "Introducción a la prospección geofísica". Cuadro comparativo de los principales métodos de prospección. Capítulo 1. p.p. 16-17. Ver bibliografía al final de la tesis.

---

## Capítulo 2

# El método de prospección sismológica de reflexión

Los métodos sísmicos son la técnica más usada de entre todas las de prospección geofísica. En él, el geofísico se sirve de una propiedad de las rocas: la de transmitir (más o menos deprisa, según el tipo de roca) las vibraciones. Esta propiedad se ha determinado de antemano para cada muestra en pruebas de laboratorio.

El método consiste en producir un microterremoto mediante la explosión de una carga de dinamita. En los alrededores del punto de explosión se colocan unos pequeños sismógrafos que registran la sacudida en diferentes intervalos de tiempo, según estén más o menos cerca de la explosión. El choque propiamente dicho va siempre precedido o seguido de otras sacudidas, producidas por la reflexión de la onda de choque en las estructuras geológicas. Reuniendo y comparando los registros de todos los sismógrafos se suelen tener suficientes indicaciones para reconstruir la geología del subsuelo.

Posteriormente viene la fase más delicada: la interpretación de los resultados. En esta hay que adivinar cuál es la estructura geológica en profundidad sin contradecirse con la disposición de las rocas de la superficie ni con los registros obtenidos por los sismógrafos.

El método de reflexión sísmica es el que aporta un cuadro más directo y detallado de la estructura geológica del subsuelo. Una ventaja singular del método de reflexión consiste en que permite levantar el mapa de muchos horizontes desde cada punto de explosión. La precisión del mapa obtenido es la misma, aproximadamente, para los horizontes más profundos que para los más someros; en cambio, en todos los restantes métodos geofísicos la seguridad decrece al ir aumentando la profundidad de penetración.

En muchos casos las exploraciones por reflexión van precedidas de reconocimientos menos costosos hechos con técnicas de gravedad, magnéticas o de refracción, para poder limitar el campo de reflexión a las áreas que ofrezcan particular interés. Los métodos de reflexión son empleados casi exclusivamente para la prospección petrolífera, dado que no son aplicables a las profundidades pequeñas a las que, de ordinario, se buscan las menas minerales; donde da los mejores resultados es en aquellas áreas donde el petróleo se encuentra retenido en trampas estructurales, pero algunas veces es útil para localizar y detallar ciertos tipos de rasgos estratigráficos. Aunque a la prospección por reflexión se le deben descubrimientos en la mayoría de las comarcas petrolíferas del mundo, existen algunas zonas en donde resulta muy difícil y costoso obtener reflexiones que puedan ser aprovechadas, debido, en general, a interferencias de ruidos.

## 2.1 Instrumentos utilizados en los trabajos de prospección sísmica.

Las técnicas y los aparatos empleados en los trabajos de exploración con los métodos sísmicos han sido perfeccionados de continuo desde que se empezaron a utilizar a principios del presente siglo, en especial a partir de la Segunda Guerra Mundial debido en gran parte al ritmo acelerado de la actividad explorativa en esta fase de la industria petrolífera. Los instrumentos utilizados por los métodos sísmicos de refracción y reflexión son casi idénticos en su fundamento y disposición. A pesar de que en los procedimientos de campo empleados para ambas técnicas existen algunas diferencias, los dos tipos de exploraciones se llevan a cabo de manera esencialmente similar. A continuación se describen brevemente estos instrumentos.

**Geófonos.** El geófono, llamado también detector, sismómetro o hidrófono en el caso de exploraciones marinas, transforma las oscilaciones sísmicas del suelo en señales eléctricas. En la exploración de petróleo, los instrumentos usados de ordinario sólo son sensibles a vibraciones de entre 5 y 100 ciclos por segundo (cps) para los trabajos de refracción, y de entre 10 y 150 cps para los de reflexión. En ambas aplicaciones el geófono sólo responde a la componente vertical del movimiento del suelo. Para los fines prácticos se considera que el geófono se mueve como si formara parte del suelo, bien como enterrado o simplemente depositado sobre la superficie. Todos los tipos modernos requieren amplificación y registro eléctricos, ya que se espera que movimientos del suelo de un orden tan pequeño como  $10^{-8}$  cm den inflexiones observables en los registros.

Existen diferentes tipos de geófonos de acuerdo a la tecnología empleada en su construcción, la figura 2.1 muestra las más representativas en la construcción de estos instrumentos.



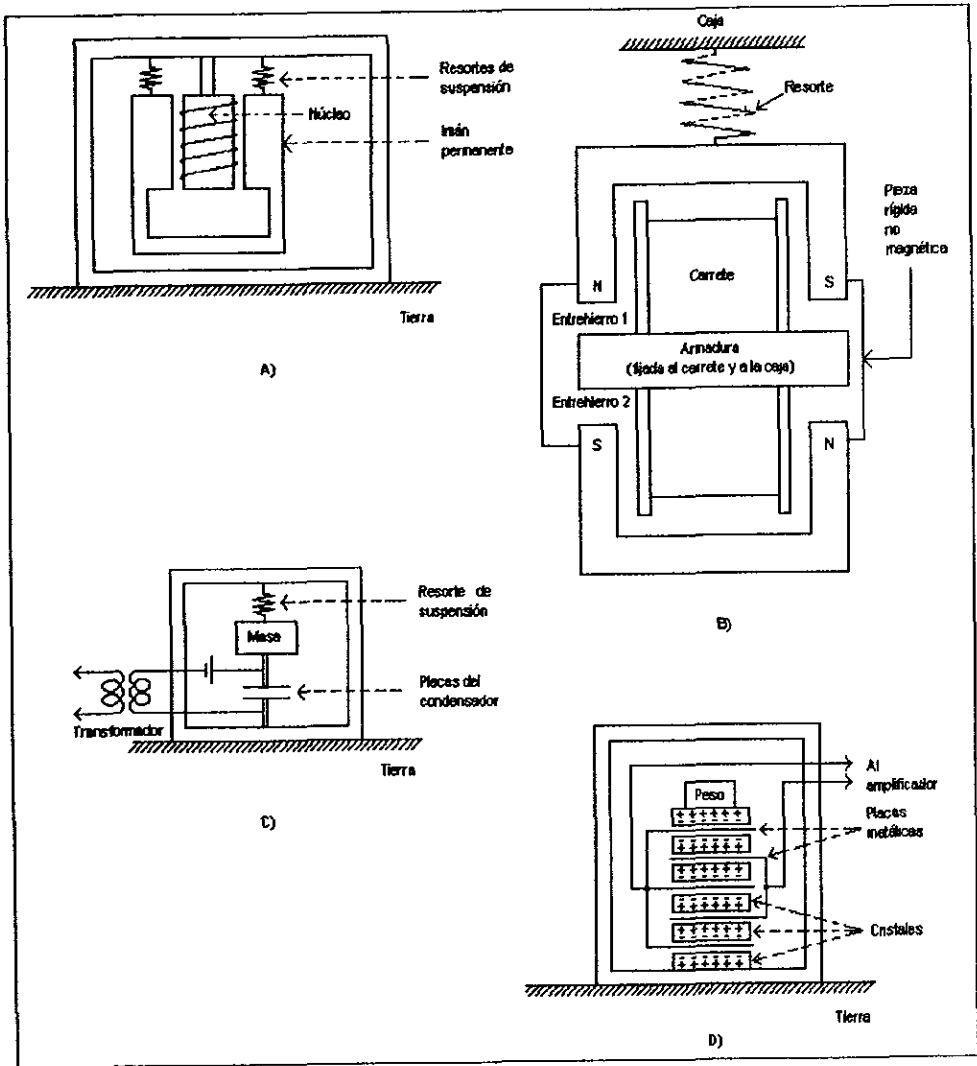


Figura 2.1. En esta serie de diagramas se muestra la estructura interna de diferentes tipos de geófonos empleados en los trabajos de prospección sísmica. A) Geófono electromagnético. El imán es el elemento inerte, la caja se mueve con el suelo. B) Esquema del geófono de reluctancia variable. C) Geófono de capacidad. D) Geófono piezoeléctrico.

**Amplificadores y filtros.** Todos los sistemas de sismógrafos comerciales emplean amplificadores electrónicos entre los geófonos y las unidades registradoras cuyos circuitos varían mucho en su disposición; algunos están acoplados por resistencia, otros por impedancia y transformador.

Los requisitos de los amplificadores sísmicos son más exigentes que los de los amplificadores construidos para la mayoría de otros fines como por ejemplo amplificadores de audio. Debido a las variaciones en el nivel del movimiento del suelo, hay que manejar voltajes que cubran amplitudes en un ámbito de 20 000 : 1, si bien los oscilógrafos registradores sólo pueden resolver diferencias del orden de 10 : 1. Por lo tanto, hay que dotar al sistema de una relación de amplificación variable.

Todos los circuitos están provistos de una serie de filtros para regular las características de frecuencia del sistema registrador. En general, es conveniente eliminar las frecuencias bajas con el fin de impedir que las ondas superficiales y los ruidos de otro tipo interfieran con las reflexiones. De modo análogo, las frecuencias elevadas son atenuadas con el fin de eliminar el ruido del viento, el del agujero y otros efectos extraños. Las frecuencias altas y bajas de corte que dan reflexiones de mejor calidad varían mucho de un lugar a otro, dependiendo de las características de frecuencia, de las señales de las reflexiones y del ruido en cada emplazamiento. Por esta razón se dispone de un amplio equipo de combinaciones de filtros de paso alto y bajo.

De igual forma, todos los circuitos amplificadores usados en los trabajos por reflexión y en los de segundos acontecimientos por refracción están dotados de un amplificador que aumenta la amplificación a medida que aumentan los tiempos en el registro, lo cual es necesario porque los acontecimientos tardíos representan ondas que han recorrido mayores distancias a través del suelo y, por lo tanto, están mucho más amortiguados que las ondas que llegan a la superficie después de haberse reflejado en horizontes de poca profundidad.

**Elementos oscilográficos y cámaras.** Los primeros oscilógrafos empleados en la prospección sísmica eran del tipo de galvanómetro de cadena, pero en la actualidad se usan más los galvanómetros de bobina. Como las modernas cámaras registradoras fotografían frecuentemente 24 o más trazos sobre una tira de papel de 20 cm de anchura, y como se necesita un galvanómetro para cada trazo, en la disposición de estos aparatos es de la mayor importancia que tengan un reducido volumen.

En el galvanómetro de bobina, una pequeña espira por la que pasa la corriente está suspendida entre los polos de un imán, de modo que las variaciones de la corriente hacen girar la espira. Un diminuto espejo fijado a la espira refleja un rayo luminoso sobre la tira de papel, quedando registrado fotográficamente sobre la tira de papel en movimiento el desplazamiento del espejo.

**Líneas de cronometraje.** La superposición fotográfica de las líneas de cronometraje sobre los trazos que registran los movimientos del suelo ha hecho posible la medición exacta, sobre diagramas sísmicos, de los tiempos en que aquellos han tenido lugar. Estas líneas son proyectadas a intervalos de 0.001 a 0.005 segundos y, por lo general, ocupan todo el ancho del diagrama. Un diapason sintonizado con precisión controla un oscilador que regula la velocidad de un disco giratorio accionado a motor, que lleva ranuras o radios dispuestos de tal manera que permitan que el rayo luminoso llegue a la cámara con los debidos intervalos.

En algunas cámaras, las ranuras interrumpen el rayo luminoso cada 0.01 segundos, y cada décima ranura es algo más ancha que las otras, de modo que habrá una línea más fuerte cada 0.1 segundos para facilitar el montaje de los tiempos en los diagramas. Más corrientemente, las líneas de 0.01 segundos alternan con otras más claras, lo que facilita la lectura de los acontecimientos, por interpolación, hasta la milésima de segundo más próxima.

Al reproducir las cintas magnéticas, las líneas de cronometraje en el registro final del papel son grabadas, en general, por los impulsos de tiempo que figuran en la cinta, de tal modo que el momento de la explosión está representado por una línea más gruesa cada décima de segundo, lo que facilita la lectura de los tiempos en el registro.

**Indicador del momento de la explosión.** Una de las trazas del diagrama se utiliza para indicar el momento o instante exacto del disparo, y a este fin el instrumento registrador está conectado con el lugar de la explosión por un alambre, en el caso del trabajo por reflexión, o por radio, cuando es por refracción (se suele emplear la frecuencia modulada para eliminar las interrupciones parásitas ocasionadas por las cargas estáticas). En la mayoría de los casos el galvanómetro responde a un impulso inductivo asociado a la interrupción de un circuito simultáneamente con la explosión. La precisión del momento de la explosión puede ser aumentada haciendo la señal proporcional a la velocidad de la variación de la corriente en el circuito abierto en el momento de la explosión.

**Equipos de registro magnético.** Hacia 1951 se introdujo, con base experimental, una nueva técnica para registrar en el campo los datos sísmicos. Esta técnica representaba la adición de una nueva fase al proceso corriente de registros, a saber, el almacenamiento de las señales sísmicas en forma reproducible en una cinta de plástico de canales múltiples, revestida magnéticamente, procedimiento desarrollado después de la Segunda Guerra Mundial para registrar sonidos y otros fines. La ventaja de este método residía en el hecho de que los datos que cubrían toda la gama de frecuencias sísmicas podían ser almacenados en la cinta en el momento del registro original, y los registros de papel empleados para la interpretación final podían obtenerse reproduciendo la cinta a través de los filtros que se deseasen.

En los registros corrientes obtenidos en el campo sobre papel no es posible comparar los efectos de los diferentes montajes de filtros en el punto de explosión sin provocar explosiones separadas para cada montaje. Con frecuencia, los pozos de explosión colapsan después de una sola explosión, y un segundo tiro supone tiempo y el gasto de otra perforación. Aun cuando el pozo no quede realmente inservible después de una explosión, no es posible, en general, colocar una segunda carga en el pozo a la misma profundidad que la primera, y un cambio en la profundidad de la explosión puede dar por resultado una reflexión de carácter totalmente diferente, aun con el mismo montaje de filtros. Por lo tanto, es difícil aislar el efecto del filtrado sólo con las técnicas corrientes de registro.

Durante los años que siguieron a la introducción de este método, el registro magnético fue adoptado, cada vez con mayor frecuencia, para las operaciones de exploración sísmica, y para 1956 ya habían sido construidas más de la mitad de las unidades registradoras magnéticas en comparación al número de equipos sísmicos que actuaban. En 1959, las tres cuartas partes, por lo menos, de todos los equipos usaban ya el registro magnético.

Aunque en la prospección sísmica se usan diversos tipos de sistemas magnéticos, las únicas diferencias realmente importantes son las que existen entre los que emplean la modulación de la frecuencia y los que emplean la modulación de la amplitud. En el primer tipo, la señal de salida del amplificador sísmico para a un modulador en el cual una señal portadora de amplitud constante, que tiene una frecuencia próxima a 300 cps, es modulada en una gama sísmica de 5 a 300 cps, por la señal recibida. La señal es extraída de la cinta, al ser reproducida, empleando un demodulador. En los sistemas de amplitud modulada la señal es grabada directamente sobre la cinta, de la misma manera que en un registro de sonido, siendo proporcional la magnetización de la cinta a la intensidad de la señal.

## 2.2 Operaciones sísmicas en tierra.

En la figura 2.2 de la página 15 se muestra un diagrama esquemático de las operaciones de prospección sísmica en tierra. La rutina de campo es a grandes rasgos como sigue: los topógrafos, por lo general, preceden unos cuantos días a los barrenos en el campo de operaciones para elegir los emplazamientos exactos para los barrenos, teniendo en cuenta factores tales como la facilidad de acceso para los camiones pesados, y la distancia a las líneas de alta tensión. Distancias y alturas han de ser medidas con gran exactitud mediante itinerarios o con la *alidada* para su subsiguiente empleo en los planos. Los barrenos, al llegar al emplazamiento, excavan los pozos para lodo, montan el tren de sondeo y durante la perforación llevan razón de todas formaciones encontradas. Los barrenos o perforaciones tienen, por lo general, una profundidad que varía entre 7.50 y 75 metros, dependiendo de la profundidad a la que se encuentra la roca consolidada y de otras condiciones locales.

Cuando las cuadrillas de dinamiteros y registradores llegan al campo de operaciones se acostumbra situar el camión registrador cerca del centro de la cadena de detectores. Los electricistas sacan el cable del tambor del camión y lo llevan hasta los emplazamientos de los detectores, que posteriormente conectan al cable mediante pinzas o enchufes. Entre tanto, los dinamiteros han estacionado su camión a corta distancia del barreno, lo cargan con cartuchos de dinamita y conectan el explosor situado en el camión con un fulminante contenido en uno de los cartuchos. El observador comprueba el circuito de sus detectores, pone en marcha la tira de papel en su cámara oscilográfica y da entonces la señal para estallar la carga. La señal se da por teléfono en el trabajo por reflexión, y por radio en el de refracción.

Una vez aquietados los trazos se para la cámara, y el registro, una vez revelado, es examinado por el observador. Cuando es necesario se disponen el montaje de los instrumentos y el tamaño de la carga para repetir la explosión con el mismo montaje detector. Aun empleando la detección magnética, conviene a veces registrar las explosiones a varias profundidades en el mismo barreno.

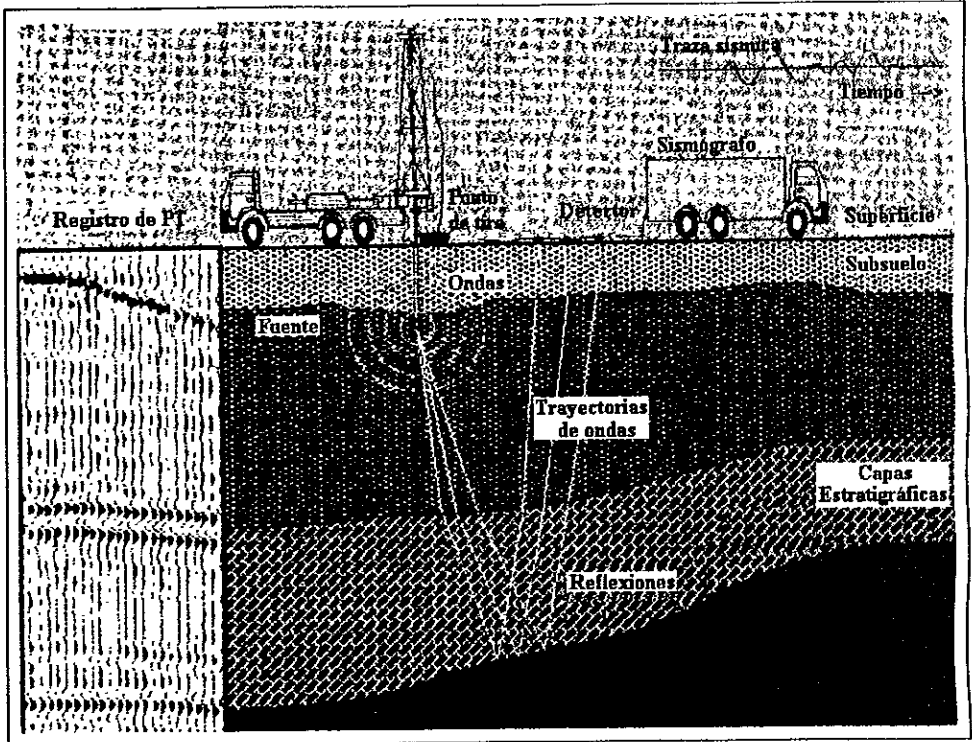


Figura 2.2. Diagrama esquemático de los trabajos de prospección terrestre. El equipo normal está formado por 15 a 20 hombres, siendo imprescindible la participación de las siguientes personas: el *jefe de equipo*, que inspecciona el personal y en muchos casos está encargado de la interpretación; el *calculador*, con uno o dos ayudantes, hace el trabajo de rutina de transformar los registros de reflexión dándoles la forma en que son usadas finalmente; el *observador*, u operador está encargado del equipo de registro, incluyendo los barreneros y los electricistas, el *topógrafo*, y su ayudante disponen los perfiles de explosión de acuerdo con las instrucciones del jefe del equipo, teniendo que marcar todos los puntos de explosión y las estaciones de detección, los *electricistas*, colocan los detectores sobre el suelo y los devuelven después de cada explosión a un camión especial y por último el *dinamitero*, que con su ayudante coloca la dinamita en el barreno mediante pértigas y hace detonar la carga con un explosor cuando así se lo ordena por teléfono el observador.

### 2.3 Operaciones en zonas cubiertas por el agua.

Esto se ilustra en la figura 2.3. A grandes rasgos los trabajos de prospección sísmica marina se pueden dividir en dos grandes grupos: reflexión en aguas someras (de poca profundidad) y operaciones mar adentro sobre las plataformas continentales. En la primera de ellas, los equipos de exploración trabajan desde tripodes montados en el agua y, en algunos casos, con agua hasta la cintura. Los hoyos para las explosiones son hechos a chorro, debajo del agua, con torres de perforación montadas en embarcaciones. La carga es introducida en los hoyos y hecha estallar desde una embarcación. La embarcación portadora de los instrumentos corresponde al camión registrador de las operaciones en tierra firme y contiene, en general, la misma clase de equipo. La comunicación se establece siempre por radio.

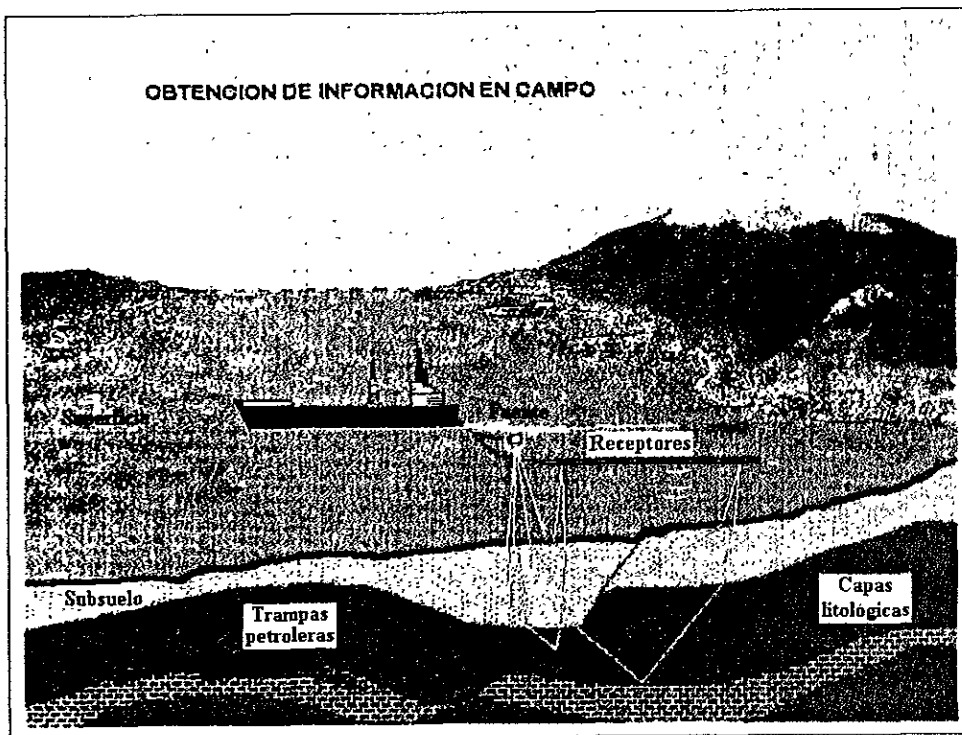


Figura 2.3. En los trabajos de prospección marina participan de ordinario tres embarcaciones: un *barco dinamitero*, cuya función es la de perforar los hoyos, introducir y detonar la carga explosiva, un *barco registrador*, que lleva los instrumentos de medición a bordo, corresponde al camión registrador en las operaciones en tierra firme y contiene, en general, la misma clase de equipo. Esta embarcación arrastra todo el conjunto de detectores por el fondo del mar, de una estación detectora a otra. La serie de detectores tiene normalmente de 150 a 300 metros de longitud, con una separación entre detectores de 15 a 30 metros. Por último, un *barco de reconocimiento* se encarga de indicar los puntos de detección por medio de globos. La comunicación se establece siempre por radio.

Por otra parte, para los trabajos de prospección mar adentro se plantean nuevos problemas que conducen a muchas técnicas operatorias nuevas. Para las investigaciones demasiado alejadas de la orilla, que quedan fuera de la vista de puntos de referencia situados en tierra, han sido adaptadas nuevas técnicas electrónicas para localizar las posiciones, desarrolladas originalmente para fines bélicos. La construcción interna de los detectores es la misma que la de los utilizados en tierra, diferenciándose solamente en las cajas y, por lo general, llevan un montaje en serie, uno detrás de otro, para que queden debidamente nivelados y colocados durante la operación.

Todo el montaje esta acomodado en una caja de forma hidrodinámica y conectado por un cable eléctrico a la embarcación detectora. En esta disposición la embarcación registradora arrastra toda la sarta de detectores por el barro del fondo del mar, desde una estación detectora a otra. Según otro sistema, se emplean flotadores de corcho para mantener los detectores suspendidos a unos 2 metros por debajo del cable, que es retenido cerca de la superficie mediante una tabla con aletas. Este sistema es aconsejable principalmente cuando la presencia de coral en el fondo de las aguas aconseja suspender los tirantes del detector. Por supuesto, las distancias entre detectores durante las explosiones se fijan cuando el cable esta tirante.

Las cargas explosivas dejadas caer por la embarcación correspondiente quedan flotando justo por debajo de la superficie del agua, sostenidas por globos, en posiciones previamente determinadas e indicadas por boyas-globos mantenidos en su sitio por contrapesos y dejadas caer por el equipo de prospección. Cuando la embarcación registradora que arrastra los detectores llega a la boya que marca su estación, da marcha atrás para aflojar el cable del detector, mientras la embarcación destinada a producir las explosiones deja caer las cargas en la boya del lugar donde haya de ser hecho el disparo y se aleja a una distancia prudencial. Se provoca la explosión con el menor retraso posible para reducir al mínimo la deriva de los detectores. Las distancias entre detectores y explosiones son determinadas con la suficiente exactitud cronometrando en el agua la onda explosiva directa.

## 2.4 Geometría de las trayectorias de reflexión en superficies de separación horizontales e inclinadas.

En el caso de una onda explosiva producida por una detonación de dinamita, justo por debajo de la superficie del suelo, cada superficie de separación que represente un cambio en las propiedades elásticas reflejan hacia la superficie parte de la energía, y un detector situado en esta responderá a la llegada de cada onda reflejada, utilizándose el tiempo empleado en el recorrido completo, desde el punto de explosión al detector, para calcular la profundidad a que se encuentra el horizonte reflectante. En las figuras 2.4 y 2.5 se ilustra la reflexión que se produce dependiendo de si el horizonte en profundidad se encuentra horizontal o inclinado con respecto a la superficie del suelo.



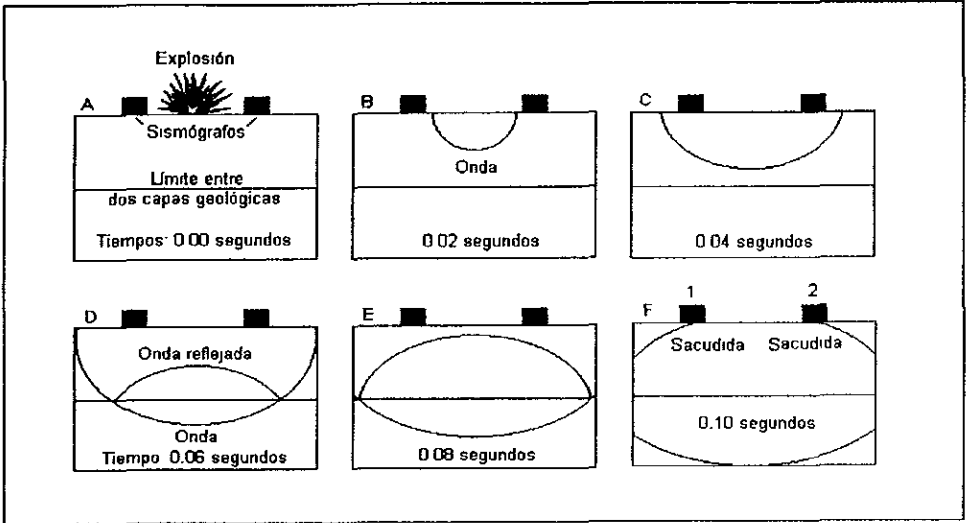


Figura 2.4. Los límites entre las capas geológicas son para las ondas como espejos en donde se reflejan. Las estructuras geológicas reflejan la onda hacia la superficie y allí es detectada por los sismógrafos. Esta secuencia de dibujos ilustra la geometría de las trayectorias de reflexión en superficies de separación horizontales. Los sismógrafos 1 y 2 muestran simultáneamente la onda (entre los dibujos B y C) y la onda reflejada (en el dibujo F)

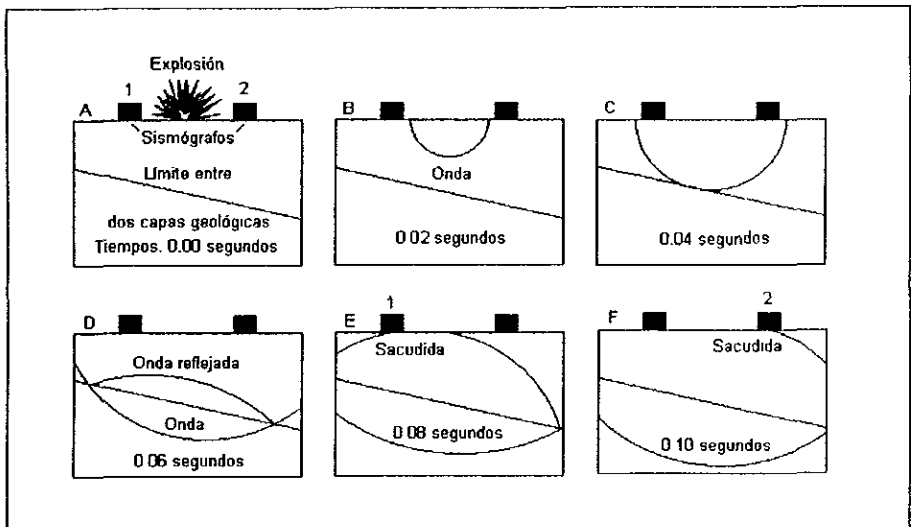


Figura 2.5. Esta otra serie ilustra el caso de un límite oblicuo. La onda también es registrada simultáneamente, pero la onda reflejada es detectada antes por el sismógrafo 1 (dibujo E) que por el sismógrafo 2 (dibujo F).

## 2.5 Diagramas y perfiles de los datos de reflexión.

Cuando se hace estallar dinamita en un pozo, las ondas registradas por los instrumentos detectores próximos habrán seguido una gran diversidad de trayectorias, cada una de las cuales requiere un tiempo de recorrido diferente, desde el punto de explosión hasta el detector. Durante varios segundos siguientes a la primera llegada de energía al detector, el suelo por debajo de este se encontrará en continuo movimiento por efecto del impacto de las ondas que se han propagado siguiendo trayectorias diversas. Algunas ondas se habrán refractado y otras se habrán dispersado; algunas se habrán propagado por la superficie del suelo y otras se habrán reflejado hacia arriba, en las distintas superficies de separación.

Para poder utilizar la energía reflejada, habrá que diferenciarla del movimiento del suelo debido a ondas de otras clases, lo que no podría lograrse empleando un solo detector. En la práctica, y para recibir las ondas de cada explosión, se hace uso de cierto número de detectores colocados a pequeños intervalos a lo largo de una línea dirigida al punto de explosión (denominada comúnmente como *línea de tiro*). El movimiento del suelo correspondiente a cada instrumento es registrado con un trazo separado en una misma banda. Las ondas correspondientes a una reflexión aparecerán alineadas en el diagrama de tal modo que las crestas o los valles de las trazas adyacentes tendrán el aspecto de encajar unas en otras. Esta alineación puede ser solamente observable para un solo ciclo de ondas en cada trazo; pero, con frecuencia, persiste durante dos o más ciclos sucesivos. El tiempo diferencial (incremento de tiempo) de las llegadas de una cresta o valle dados, a las sucesivas posiciones del detector, nos informa acerca del *buzamiento* de la capa reflectante, y el tiempo absoluto indica su profundidad con respecto a la superficie.

Cuando los datos sísmicos son registrados en cinta magnética, es posible transformar los registros en perfiles de diferente forma. El tipo más usado consiste en hacer las gráficas correspondientes únicamente al *wiggle*, como en los registros usuales, alineados en secuencias a lo largo del perfil. Existen dos variantes de este tipo: 1) el de trazos cortados, en el cual los picos y los valles que tienen una amplitud mayor que un valor predeterminado, son cortados, dejando abiertos por arriba y por abajo los ciclos más fuertes, y 2) el de trazos plegados, en el cual las partes de los trazos que se salen de los niveles límites son movidos hacia atrás, más de una vez si fuera necesario, hasta la zona entre estos niveles. Con esta disposición, las partes de mayor amplitud del ciclo se acomodan en las partes de menor amplitud. Estas dos variantes tienen la ventaja de que los trazos sucesivos no pueden cruzarse sin tener en cuenta sus amplitudes.

Otro tipo de perfil presenta cada trazo en forma de una tira de película de densidad variable, en la cual los picos del registro corriente aparecen en negro y los valles en blanco. Los niveles de amplitud intermedia están representados por varios matices de gris. También hay perfiles en forma de película de área variable, en los cuales las anchuras de las partes negras y blancas de la película son proporcionales a la intensidad de la señal. En la figura 2.6 se muestra una comparación de los diferentes tipos de perfiles descritos en los párrafos anteriores.

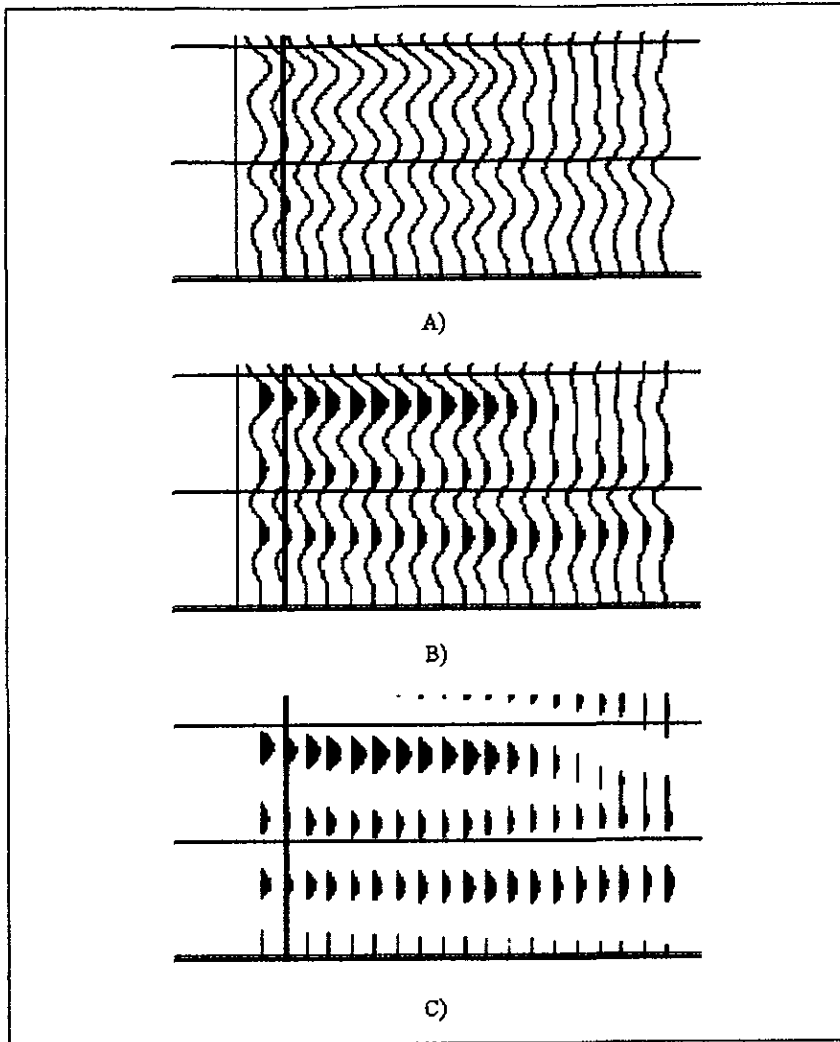


Figura 2.6. Modos alternativos de presentación del registro sísmico. A) Únicamente el *wiggle*. B) Densidad variable (área variable, *wiggle* y relleno positivo de la traza) C) Únicamente el área variable.

Todos estas modalidades en la presentación de las trazas sísmicas serán contempladas en las gráficas generadas por el sistema. Pero antes de describir el desarrollo del programa, en la segunda parte de la tesis se describirá brevemente la manera en que los datos sísmicos son analizados en la Gerencia de Prospección Geofísica del IMP, así como la forma que adoptan antes y después de ser procesados por el sistema de conversión.

---

## **Parte II**

### **Antecedentes en el desarrollo del sistema**

---

## Capítulo 3

# Secuencia de proceso de la información de campo en la Gerencia de Prospección Geofísica del Instituto Mexicano del Petróleo

El Decreto Presidencial del 23 de agosto de 1965, expedido por el Lic. Gustavo Díaz Ordaz, mediante el cual se fundó el Instituto Mexicano del Petróleo (IMP) como un organismo descentralizado, de interés público, con carácter preponderantemente técnico, con personalidad jurídica y patrimonio propio, tuvo su origen en el grado de dependencia tecnológica extranjera, que en materia petrolera era aun considerable en esa época. Esta situación, derivada del esfuerzo y atención por parte de Petróleos Mexicanos, que a raíz de la expropiación petrolera se dirigió primordialmente al suministro de los energéticos necesarios para el progreso y desarrollo del país, no era lo conveniente para el modelo de la economía nacional que la expropiación petrolera había planteado.

Los regímenes presidenciales de los licenciados Miguel Alemán, Adolfo López Mateos y Adolfo Ruiz Cortines promovieron en diferentes facetas el arranque industrial del país, impulsando la explotación de nuestras reservas petroleras y ampliando y desarrollando las instalaciones y plantas industriales de Petróleos Mexicanos como palanca para contribuir al desarrollo económico y social del país.

Desafortunadamente las políticas para sustituir importaciones de tecnología, de ingeniería, de mano de obra especializada y capacidad de construcción, no produjeron los resultados esperados y el crecimiento de la industria petrolera se vio afectado por una considerable dependencia en tecnología del extranjero.

Así en la administración del Lic. Jesús Reyes Heróles, en funciones de Director General de Pemex, se presentó la iniciativa al Ejecutivo Federal, proponiendo la creación de un organismo que apoyara a Petróleos Mexicanos en la solución de sus problemas tecnológicos y de recursos humanos.

Para asegurar los compromisos de la Institución recién creada, a fin de satisfacer las necesidades de Petróleos Mexicanos, así como su vinculación con los sectores académico y de investigación, la integración de su Consejo Directivo comprende a representantes de Pemex y de las principales instituciones de educación superior.

El inicio de su operación se propició con la participación de un grupo de profesionales destacados de Pemex que contaban, además de gran capacidad técnica, con amplio conocimiento de la industria petrolera, y con un conjunto de investigadores del sector académico con vasta experiencia en el desarrollo de proyectos de investigación científica.

De esta manera se formaron grupos de geólogos, geofísicos, matemáticos, físicos, químicos, electrónicos e ingenieros petroleros y químicos, entre otros especialistas, que en colaboración con colegas conocedores de los problemas centrales de las industrias petrolera, petroquímica y química, comenzaron a definir las áreas o temas en que era conveniente desarrollar esfuerzos para apoyar las actividades de la industria y adelantar sus posibles demandas tecnológicas.

### 3.1 Estructura orgánica del Instituto Mexicano del Petróleo.

Esta se divide en seis Subdirecciones y tres Coordinaciones, todas dependen directamente de la Dirección General. La figura 3.1 ilustra el organigrama de la Institución.

#### 3.1.1 Subdirección de Exploración y Producción.

La Subdirección de Exploración y Producción realiza actividades concernientes al petróleo. Los trabajos, servicios y productos que la subdirección proporciona a la división de Pemex Exploración-Producción (PEP), son de gran ayuda en la prospección y extracción del petróleo, reafirmando así esta subdirección como el brazo tecnológico de Pemex.

#### Exploración.

La exploración y producción de los hidrocarburos constituyen las actividades primarias de la industria petrolera nacional y son el punto de partida para su planeación integral y desarrollo. Corresponde al Instituto Mexicano del Petróleo otorgar el apoyo necesario para el cumplimiento de las funciones en materia de investigación, desarrollo tecnológico y prestación de servicios técnicos especializados.

Desde su fundación, el Instituto Mexicano del Petróleo ha concedido una importancia primordial a proyectos de investigación, desarrollo tecnológico, estudios y prestación de servicios en cada una de las dos grandes ramas en que se sustenta la exploración: la prospección geológica y la prospección geofísica, sin omitir las ciencias relacionadas con el conocimiento del subsuelo.

Entre las ciencias afines a la geología se ha conservado una posición de vanguardia en paleontología y estratigrafía, que permiten ubicar en el tiempo las secuencias acumuladoras de hidrocarburos.

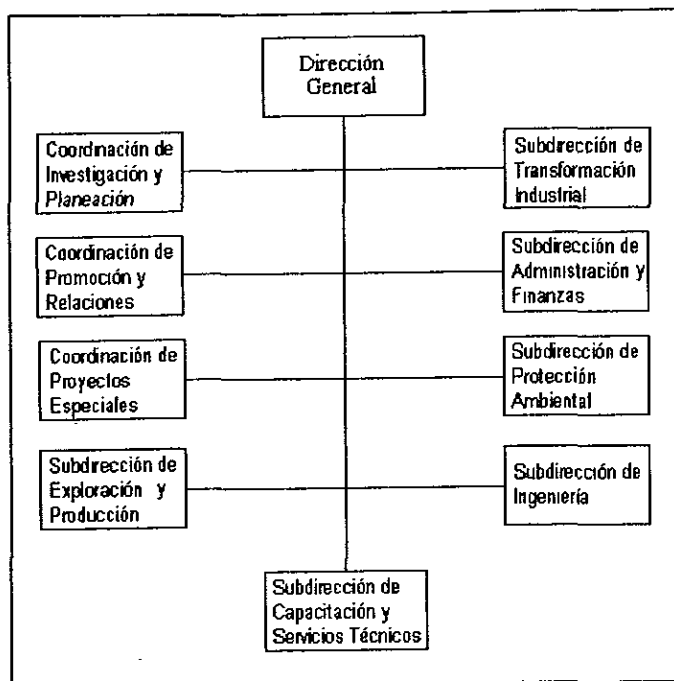


Figura 3.1. Estructura orgánica del Instituto Mexicano del Petróleo

La prospección sísmica representa el método más avanzado y preciso para la localización y cuantificación de yacimientos petroleros. El IMP procesa e interpreta la información que recibe de Petróleos Mexicanos, utilizando sistemas informáticos de gran capacidad y eficiencia.

Por otra parte, para el procesamiento sísmico, destacan el modelado tridimensional y la tomografía sísmica. Esta tecnología de punta permite determinar las formas y dimensiones de las estructuras acumuladoras de hidrocarburos y realizar secciones en cualquier dirección.

Actualmente, con los avances de las geociencias y la prospección geofísica, utilizando recursos informáticos complejos, es factible llevar a cabo la evaluación económica de las acumulaciones de petróleo y gas en el subsuelo, con un alto grado de certidumbre, con lo que se reduce significativamente el riesgo económico de las inversiones en exploración.

## Producción.

La perforación de pozos es una actividad que permite establecer el conducto para la extracción de los hidrocarburos del subsuelo. Su costo alcanza cifras del orden de varias decenas de millones de nuevos pesos, por este motivo, la tecnología para optimar la perforación requiere de información de laboratorio directa e indirecta, de la naturaleza de la roca y de los fluidos, así como del equipo de perforación, a fin de minimizar costo y tiempo del proceso dentro de los márgenes internacionales de seguridad. El IMP ha desarrollado equipos y tecnología de vanguardia para este proceso.

Durante la perforación es necesario mantener la estabilidad del pozo, lubricar adecuadamente la barrena de perforación y obtener del subsuelo las muestras de las formaciones geológicas que se atraviesan, lo cual se logra en parte con el uso de fluidos y aditivos de perforación. Es importante señalar que el IMP cuenta con los equipos de vanguardia para realizar los estudios que conlleven al desarrollo y establecimiento de normas que garanticen su calidad

Petróleos Mexicanos ha construido cavernas subterráneas para el almacenamiento estratégico del petróleo, que le permitan disponer oportunamente de los volúmenes necesarios de hidrocarburos, para cumplir con los compromisos nacionales e internacionales en los periodos de mal tiempo o casos de emergencia. Por su parte, el IMP cuenta con la tecnología para apoyar la creación de almacenamientos subterráneos, así como con equipo y personal especializado para determinar la geometría y capacidad volumétrica de estas cavernas subterráneas.

Desde su fundación, el Instituto, además de la investigación y desarrollo tecnológico en las áreas de exploración y producción, ha incursionado exitosamente en el desarrollo de aparatos, instrumentos y equipos relacionados con las distintas fases de explotación de hidrocarburos; tal es el caso de los registros geofísicos de pozo sondas para la evaluación de propiedades petrofísicas de roca y formaciones geológicas, el sistema sonar para la determinación del volumen y geometría de cavernas en domos salinos y el sistema para prueba de presión en pozos.

El IMP debe mantener una posición de vanguardia en los procesos de las diversas etapas de la vida productiva de los campos petroleros; así, al declinar la energía propia de los yacimientos, es necesario utilizar sistemas artificiales de producción. En la selección del tipo de bombeo, su diseño e instalación, se participa con Petróleos Mexicanos para optimar su funcionamiento.

El conocimiento detallado de sistema roca-fluidos que constituye el yacimiento, es indispensable para obtener el modelo preciso y confiable que permita una representación fiel de su comportamiento en la simulación bajo diversas alternativas de explotación. Para lograr una caracterización de alta calidad se utiliza información geológica, geofísica y de prueba de presión de registro de pozo en laboratorio petrofísico. El IMP está utilizando equipo altamente especializado como el tomógrafo de rayos X y el moderno sistema de análisis de presión, volumen, y temperatura.



Para lograr la máxima utilidad económica en la explotación de los yacimientos petroleros, es necesario administrarlos integralmente durante su vida productiva, lo que implica el uso de tecnología de vanguardia de las diversas ramas de la ingeniería, geología, geofísica, yacimientos, construcción, perforación, ingeniería económica, sistemas informáticos y protección ambiental. En el IMP se dispone de la tecnología y recursos humanos especializados para coadyuvar con Petróleos Mexicanos en esta importante tarea.

PEP ha establecido una separación regional de la República Mexicana en diferentes zonas para la prospección y extracción de hidrocarburos. La información de campo proveniente de cada una de estas zonas es procesada en la Gerencia de Prospección Geofísica como se describe más adelante.

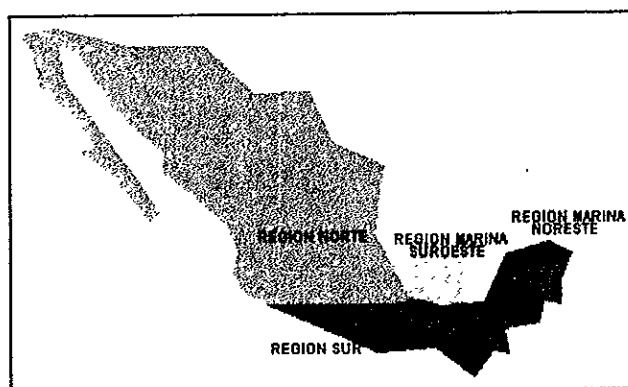


Figura 3.2. Regionalización de Pemex Exploración-Producción.

**Región Norte.** Comprende gran parte de la República Mexicana, abarcando la Península de Baja California y los estados de Sonora, Chihuahua, Nuevo León, Tamaulipas, Sinaloa, Durango, Zacatecas, San Luis Potosí, Colima, Aguascalientes, Jalisco, Nayarit, Guanajuato, Querétaro, Hidalgo, Veracruz, Michoacán, Estado de México, Morelos, Puebla, Tlaxcala y parte de los estados de Guerrero y Oaxaca. Esta región colinda al norte con los Estados Unidos de América; al sur con el paralelo 18 y el cauce del Río Papaloapan; al este con el Golfo de México y al oeste con el Océano Pacífico.

**Región Sur.** Comprende la parte media y sur de los estados de Guerrero y Oaxaca, además de Chiapas, Tabasco, Campeche, Yucatán, Quintana Roo, el Golfo de Tehuantepec y el área marina de Coatzacoalcos. Sus límites geográficos son al norte con el Golfo de México y el paralelo 18; al sur y al oeste limita con el Océano Pacífico y al este con las Repúblicas de Guatemala y Belice.

**Región marina Noreste.** Se localiza al este del meridiano 92 en la sonda marina de Campeche, dentro de la plataforma continental que rodea a los estados de Campeche y Yucatán y su cabecera se ubica en Cd. del Carmen, Campeche.

**Región marina Suroeste.** Geográficamente se ubica entre el paralelo 23 y los meridianos 92 y 95 en las costas de los estados de Tabasco y parte de Veracruz.

En la siguiente figura se ilustra el organigrama de la Subdirección de Exploración y Producción.

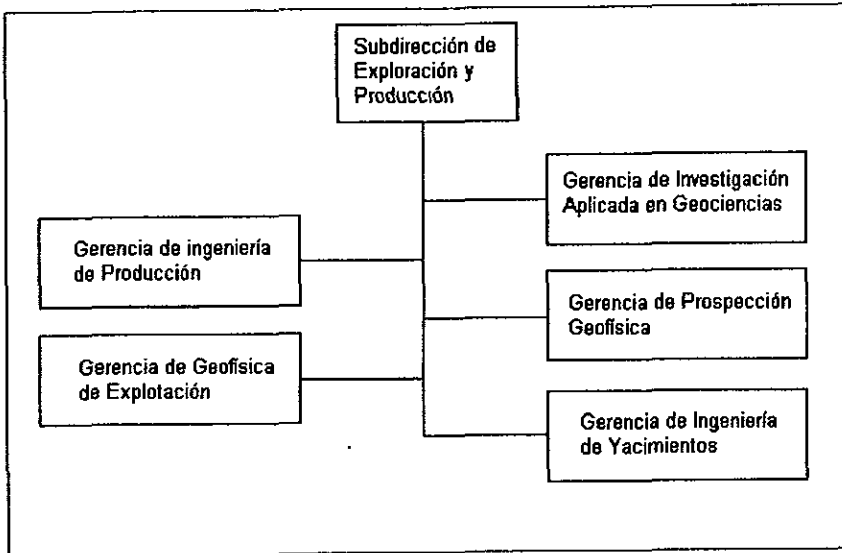


Figura 3.3. Organigrama de la Subdirección de Exploración y Producción.

### 3.1.2 Gerencia de Prospección Geofísica.

La Gerencia de Prospección Geofísica (GPG) esta dedicada a realizar investigación aplicada, desarrollo tecnológico, estudios y servicios especializados en las ramas de la prospección geofísica, en el ámbito nacional e internacional para la exploración del petróleo y otros recursos del subsuelo.

La política que se sigue dentro de la Gerencia es la conservar un alto nivel de competitividad de los productos y servicios especializados. Mantener en constante formación, capacitación y actualización al personal, así como promover y estimular la formación y desarrollo de nuevos investigadores.

Esta Gerencia se subdivide a su vez en tres Líneas de Negocios (ver Figura 3.3), para las cuales el Centro de Procesamiento Geofísico sirve de auxiliar en el proceso de la información sísmica de campo.

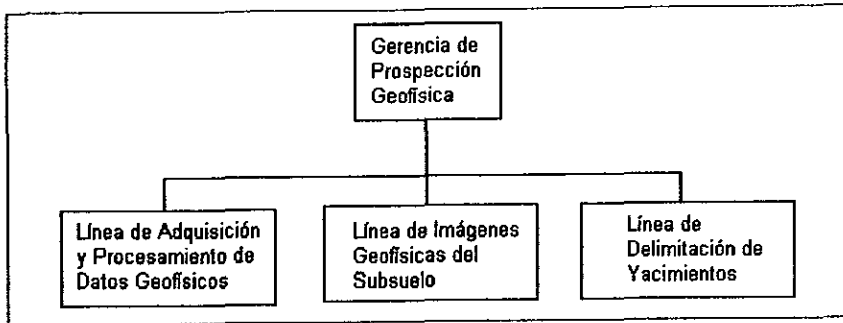


Figura 3.4. Organigrama de la Gerencia de Prospección Geofísica.

### Línea de Adquisición y Procesamiento de Datos Geofísicos.

Se encarga de prestar servicios de procesamiento de datos sísmológicos para intérpretes de las regiones de Pemex. Su enfoque primario es hacia el proceso convencional bidimensional, procesos sísmicos especiales y el proceso tridimensional. Recientemente se trabaja en proyectos como el de la unión de prospectos 3D independientes para conformar un cubo sísmico nuevo. Es en esta parte donde se producen las gráficas en dos dimensiones que serán procesadas por el sistema a desarrollar en la presente tesis.

En la Figura 3.4 se ilustra la secuencia de proceso para la obtención de las gráficas de secciones sísmicas. Cabe aclarar que hoy en día sólo se procesa información sísmica digital. La parte correspondiente al dibujo y armado de secciones se ha automatizado en los últimos años. Utilizando la computadora principal IBM 9121, esta tarea se realiza por medio de un programa de procesamiento y graficado de información sísmica denominado CAPS (Computer Aided Plotting System: sistema de graficación asistido por computadora), el resultado final se imprime en un graficador electrostático Xerox Versatec 6410 para su posterior envío a Pemex. La parte correspondiente al control de calidad es llevada a cabo por un jefe de grupo de analistas geofísicos, dependiendo de la zona que estén trabajando. Actualmente se procesa información de campo proveniente de las zonas Norte, Sur de la República y de las regiones marinas suroeste y noreste del Golfo de México (ver figura 3.2).

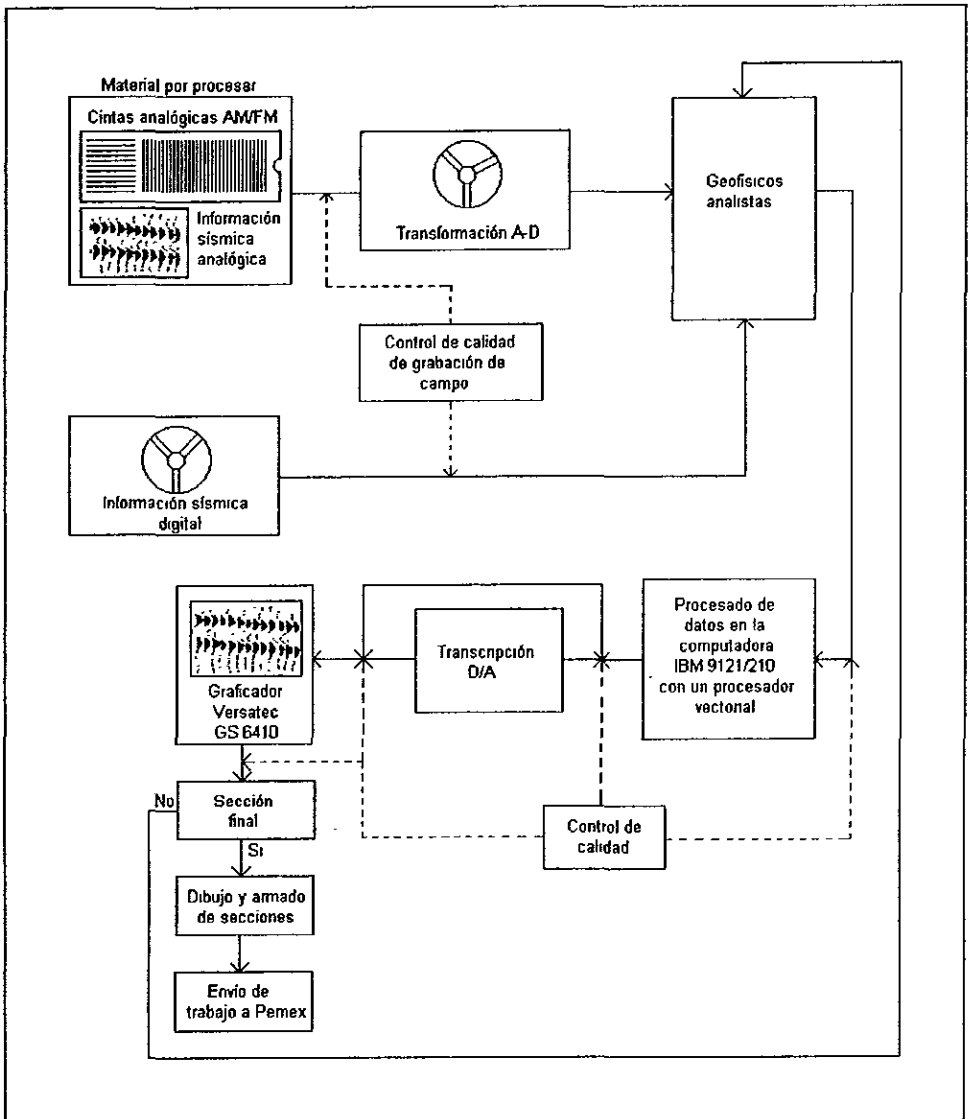
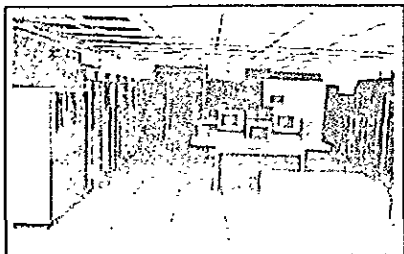


Figura 3.5. Secuencia de proceso de la información sísmica de campo en la Gerencia de Prospección Geofísica

### 3.2 El Centro de Procesamiento Geofísico.

El centro de Procesamiento Geofísico sirve de apoyo en la prestación de servicios de procesamiento geofísico y sísmico para Pemex, así como en la investigación y desarrollo de nuevas tecnologías.



Se cuenta con una computadora vectorial IBM 9121 modelo 210 con lectoras de cintas de media pulgada y cartuchos tipo 3480 y 3490, 2 unidades de disco, un graficador Xerox Versatec 6410 y 43 terminales alfanuméricas. Una computadora para proceso en paralelo IBM SP/2 9076 con 4 nodos tipo 590, 3 estaciones de trabajo, 10 terminales gráficas y 30 computadoras personales, todo esto interconectado con redes Ethernet y Token Ring.

La carga de trabajo del proceso sísmico se desarrolla con cuatro grupos de analistas geofísicos, contando con el apoyo del departamento de programación y sistemas, así como del personal en graficación, cintoteca y de operación. En las tareas de procesamiento se contempla el trabajo sísmico convencional bidimensional, proceso tridimensional y procesos sísmicos especiales (atributos, pseudovelocidades, etc.)

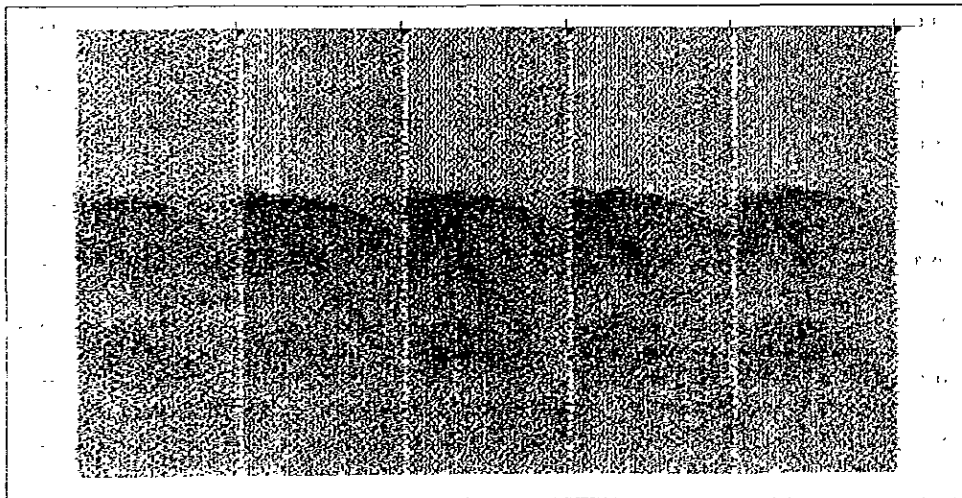
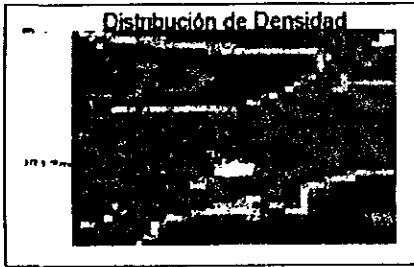


Figura 3.6. Ejemplo de una sección sísmica bidimensional



En la Gerencia de Prospección Geofísica se realiza también investigación en técnicas de adquisición y procesado de datos, así como en el desarrollo de nuevas técnicas en apoyo a la interpretación sísmica. Se trabaja en proyectos como modelado sísmico, inversión petrofísica, tomografía sísmica, proceso *subsal*, migración en profundidad y desarrollo de nuevas técnicas en la atenuación de ruidos.

Por último, se crearon, mantienen y salvaguardan bases de datos gravimétricas y magnetométricas, incluyendo además estudios geofísicos para la exploración del petróleo.



---

## Capítulo 4

### Descripción del sistema rasterizador HSR-11B

El sistema HSR-11B (High Speed Rasterizer: rasterizador de alta velocidad), es una computadora especialmente diseñada para procesar información gráfica vectorizada con el objetivo de convertirla en imágenes tipo *raster*, las cuales pueden ser visualizadas o impresas en una amplia variedad de dispositivos gráficos de salida. Este sistema opera en unión con una computadora anfitriona (*host computer*) a la cual releva del intenso proceso de manipulación de bits y operaciones lógicas asociadas a la tarea de rasterizar grandes cantidades de datos.

El hardware de esta computadora está basado en la familia de dispositivos bipolares LSI de la Advanced Micro Devices A 2900, los cuales han sido combinados para formar una sistema de alta velocidad, cuyas capacidades de rasterización abarcan no solo información gráfica vectorizada sino también datos de registros sísmicos, por lo que este sistema es muy utilizado en centros de procesamiento de información geofísica como un auxiliar para el graficado de secciones sísmicas. Los principales elementos utilizados dentro del sistema incluyen:

AM2901A	Bit slice procesor
AM 2902	High speed look ahead carry generator
AM2911	Microprogram sequencer
TDC 1010J	16 x 16 LSI Multiplier accumulator

Los anteriores componentes constituyen el núcleo principal de una computadora de 16 bits con una palabra (word) de microinstrucción de 2 bytes de longitud. Estas instrucciones se encuentran almacenadas en un chip EPROM 2716 o 2732, el cual proporciona un arreglo de 1 ó 2K palabras respectivamente. El sistema cuenta además con dos tarjetas de memoria de acceso aleatorio, la primera contiene 128K palabras de 16 bits de longitud de memoria RAM dinámica. Esta es utilizada para el almacenamiento de la imagen e incluye además 8K palabras de memoria RAM estática para almacenamiento temporal de alta velocidad. La segunda opción contiene 512K palabras de memoria dinámica y 8K palabras de memoria RAM estática.

Un conjunto de comandos bastante flexibles han sido incorporados para controlar la adquisición de datos y el procesamiento de funciones desde la computadora anfitriona. Adicionalmente, varias rutinas de autodiagnóstico han sido incluidas con el propósito de verificar las secciones del CPU y memoria del sistema, y para la creación de diversos patrones de prueba para graficadores o pantallas de visualización.

Otra parte importante de este sistema es un puerto especial para el intercambio de información gráfica con el exterior denominado Universal Graphic Bus. Este puerto y su correspondiente conjunto de tarjetas de entrada/salida proveen de una interfaz para el enlace con la computadora principal así como con otros equipos periféricos.

La siguiente figura muestra un diagrama de la ubicación del sistema rasterizador HSR-11B dentro del ámbito del ordenador principal IBM 9121.

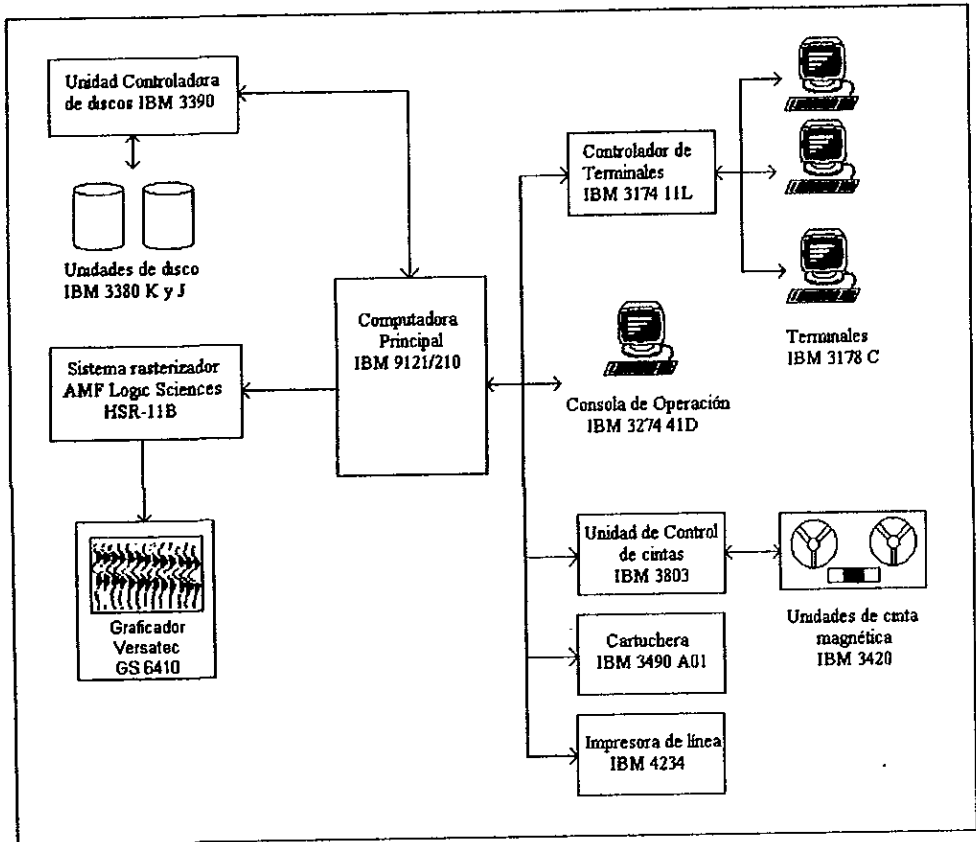


Figura 4.1. Entorno del sistema rasterizador HSR-11B. En esta figura se observa que esta computadora opera en conjunto con el ordenador principal para llevar a cabo la tarea de rasterización de datos sin que el rendimiento global del sistema se vea afectado por ello, ya que es este sistema y no la computadora anfitriona la que se encarga de dicho proceso.



## 4.1 Formato general de los comandos.

Los comandos del sistema HSR-11B siguen un formato específico. Cada orden, junto con sus correspondientes parámetros, están almacenados en palabras de 16 bits de longitud. Su forma general dentro del flujo de datos de entrada es como sigue:

Palabra (word)	Nombre
1	command type
2	word count
3	word (1)
4	word (2)
5	word (3)
:	
N	word (N-1)

La primera palabra siempre se refiere al tipo de comando a procesar por el sistema rasterizador. Estos pueden adoptar los valores del 0 al 11 y cada uno tiene un significado específico. Todos aquellos valores fuera del rango anterior no serán procesados y en su momento podrían activar una bandera de comando ilegal dentro del sistema.

Algunos comandos consisten únicamente de un solo valor, pero en la mayoría de los casos, la segunda palabra (word count), siempre se refiere al número de palabras que siguen a la orden, el *word count* está incluido dentro de este número. Todos los valores subsecuentes hasta la palabra N-1 contienen información o datos específicos para el comando que se este procesando. La siguiente tabla muestra una relación de los doce comandos que procesa el sistema HSR-11B. De todos ellos, los referentes a la información sísmica (Seismic Traces) y a la información vectorial (Sectored Vectors) de la gráfica, serán los principales elementos a procesar por el programa de conversión al momento de construir la gráfica CGM.

Tipo de comando	Función
0	No operation
1	Seismic trace
2	Timing line
3	Serial data
4	Seismic flush
5	Restore raster buffer
6	Graphic flush
7	Sectored vectors
8	Scan line advance
9	Initialize
10	Alphanumerics
11	Resampling

## 4.2 Definición de los comandos

### 4.2.1 No operation.

Palabra (word)	Nombre	Significado	Rango de valores
1	No operation	No se ejecuta ninguna tarea por parte del rasterizador	0

Este comando será ignorado por el sistema HSR-11B.

## 4.2.2 Seismic traces.

Palabra (word)	Nombre	Significado	Rango de valores
1	Line thickness and command type	Byte alto: ancho de línea Byte bajo: código del comando Seismic Trace	0 - 15  1
2	Word count type	Número de palabras que siguen al comando	
3	Display mode	Formato del dispositivo de visualización para las trazas	0-4
4	Input format	Formato de entrada de la información sísmica	0-4
5	Interpolation	Factor de interpolación lineal	1/4096 a (15+4095)/4096
6	-----	No utilizado	
7	Trace scalar word 1	Factor de escala aplicable a la amplitud de traza	Ver notas
8	Trace scalar word 2	Factor de escala aplicable a la amplitud de traza	Ver notas
9	-----	No utilizado	
10	+ Max	Máxima deflexión positiva de la traza	0-4096
11	- Max	Máxima deflexión negativa de la traza	0-4095
12	X Sstart	Inicio de la traza en el eje X	0-65535
13	Y Start	Inicio de la traza en el eje Y	0-65535
14	Fill	Area de relleno	0±4096
15	Sample 1	Muestra 1 de la traza sísmica	-128 a 127

Palabra (word)	Nombre	Significado	Rango de valores
16	Sample 2	Muestra 2 de la traza sísmica	-128 a 127 +255/256
	:		
	:		
	Sample N		

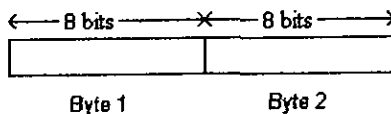
Este comando hace que el sistema HSR-11B genere una imagen rasterizada de las trazas sísmicas de entrada, con el origen de la primer muestra situado en el punto de coordenadas X Start, Y Start (Ver ejemplo más adelante).

La dirección en la cual se dibujarán las trazas sísmicas dependerá de cómo se haya establecido el valor del parámetro *seismic plot direction* dentro del comando *initialize* (ver más adelante en el comando número 9). El incremento de Y entre cada muestra dependerá del valor de interpolación (*interpolation*) y, si es utilizado, del remuestreo (*resampling*, ver comando número 11).

El parámetro *line thickness* especifica el ancho de línea en pixeles que será utilizado para el trazado del *wiggle* correspondiente de la traza sísmica. Un valor de cero o uno implica el utilizar un ancho de línea de un punto para dibujar este elemento.

Cualquier parte de la traza sísmica que no quede dentro de las líneas que se encuentren almacenadas en la memoria del HSR-11B, serán "recortadas" y no se dibujarán en dicha sección de la gráfica. Una misma traza puede estar repetida para otras secciones de la gráfica, de manera tal que aún cuando la traza tenga que ser "dividida" y transportada en varios *buffers* hacia el graficador, en este último será dibujada correctamente.

El valor de *input format* define el formato de los datos sísmicos de entrada. Por ejemplo, un valor de 1 se refiere a un formato para cada muestra de 16 bits de longitud (dos bytes), siendo el primer byte en complemento a dos y con un punto binario después de los primeros 8 bits:

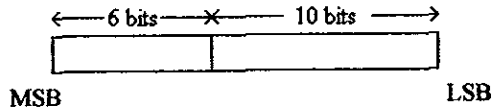


(Existe un punto binario entre ambos bytes)

Los siguientes formatos son soportados por el sistema HSR-11B:

Valor del parámetro <i>input format</i>	Formato de las muestras
0	8 bits en complemento a dos
1	16 bits en complemento a dos
2	No se utiliza ningún formato
3	punto flotante de 32 bits de IBM
4	4 bits en complemento a dos

El factor de interpolación lineal (*interpolation*) es un número real de 16 bits de longitud con un punto binario ubicado antes de los cuatro bits más significativos. El siguiente diagrama ilustra la forma en que se interpreta este valor:



El factor de interpolación lineal se define de la siguiente manera:

$$\text{Interpolación} = \frac{\# \text{ de muestras de salida}}{\# \text{ de muestras de entrada} \times \text{razón de remuestreo}}$$

donde, la razón de remuestreo (*resampling*) es igual al número de puntos generados para cada muestra cuando la función de remuestreo definida por el comando *resampling* (ver más adelante comando número 11) es aplicada a la traza. Si ninguna función de remuestreo es dada, entonces la razón de remuestreo es igual a uno.

El parámetro *display mode* define el formato de salida para las trazas sísmicas. Los siguientes formatos son soportados por el HSR-11B.

Valor del parámetro <i>display format</i>	Formato de salida
0	área variable, relleno positivo
1	área variable, relleno negativo
2	únicamente el <i>wiggle</i>
3	área variable, <i>wiggle</i> , relleno positivo
4	área variable, <i>wiggle</i> , relleno negativo

El valor *trace scalar* es un parámetro de escala aplicado a cada una de las muestras de las trazas sísmicas, antes de que sea aplicada cualquier operación de interpolación o remuestreo. El formato en el cual se presenta este valor en el flujo de datos de entrada depende directamente del parámetro *input format*:

Valor del parámetro <i>input format</i>	Trace scalar 1	Trace scalar 2
0	No utilizado	16 bits en complemento a dos
1	No utilizado	16 bits en complemento a dos
2	----- 32 bits en complemento a dos -----	
3	----- punto flotante de 32 bits de IBM -----	

Por último, el valor *fill* define la distancia sobre el eje X a partir de la cual se hará el relleno de uno de los lóbulos de la traza sísmica. Este puede ser hacia la parte positiva o negativa del *wiggle*, iniciando a partir del punto cero relativo al origen de la traza, o bien a partir de cualquier otro valor. Por convención, se asume que la parte positiva se encuentra hacia arriba, si la traza es dibujada horizontalmente, o hacia la derecha si se dibuja en forma vertical.

## Ejemplo de un bloque de traza sísmica.

Palabra	Valor	Significado
1	1	Tipo de comando ( <i>seismic mode</i> en ese caso).
2	25	Numero de palabras que siguen al comando.
3	3	Formato de salida de la traza sísmica en área variable, <i>wiggle</i> y con relleno de la parte positiva.
4	1	Las muestras de entrada están en formato de 16 bits en complemento a dos (ver párrafos anteriores).
5	1000 HEX	Factor de interpolación lineal = 1.
6	---	No utilizado.
7	---	No utilizado.
8	0400 HEX	Parámetro <i>trace scalar</i> = 1.
9	---	No utilizado.
10	50	Máxima deflexión positiva de la traza.
11	50	Máxima deflexión negativa de la traza.
12	50	Inicio de la traza en X.
13	1	Inicio de la traza en Y.
14	1	Punto de inicio del relleno = +1.
15	0000 HEX	Valor de la muestra 1 = 0
16	FF00 HEX	Valor de la muestra 2 = -1
17	FE00 HEX	Valor de la muestra 3 = -2
18	FE00 HEX	Valor de la muestra 4 = -2
19	FF00 HEX	Valor de la muestra 5 = -1
20	FF00 HEX	Valor de la muestra 6 = -1
21	0000 HEX	Valor de la muestra 7 = 0
22	0100 HEX	Valor de la muestra 8 = 1
23	0300 HEX	Valor de la muestra 9 = 3
24	0300 HEX	Valor de la muestra 10 = 3
25	0000 HEX	Valor de la muestra 11 = 0
26	0000 HEX	Valor de la muestra 12 = 0

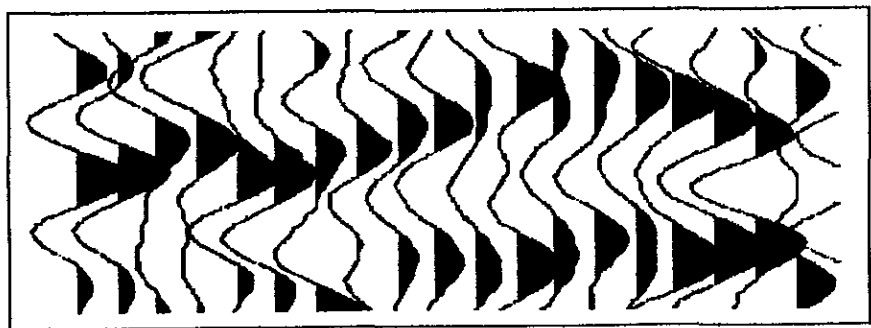
Ejemplo de la visualización de una traza sísmica (para los datos del ejemplo anterior).

El avance del papel se supone hacia arriba

	Líneas de la imagen	Valores de cada muestra
Eje X	47 . . . . .	-3
↓	48 . . . * *	-2
X Start = 50	49 . . * . . * *	-1
(punto cero de la traza)	50 . * . . . * . . * *	0
Y Start = 1	51 . . . . . * + + . . .	1
	52 . . . . . + + . . .	2
	53 . . . . . * * . . .	3
	54 . . . . . . . . . .	4
	0 1 2 3 4 5 6 7 8 9	
	Eje Y →	

De la anterior gráfica se observa que la distancia X en puntos de la envolvente de la traza sísmica es igual al valor que se obtiene de los datos de entrada para cada punto sobre el eje Y. Así por ejemplo, en la anterior figura, la tercer muestra tiene un valor de -2, por lo tanto el punto de coordenadas P(X Start - 2, 3) es encendido, es decir, un pixel es dibujado en esa posición. Por otra parte, de los datos de entrada se observa que el valor *fill* es igual a +1, con un relleno de la parte positiva de la traza. Así, se observa que el noveno y décimo valores sobre el eje Y son afectados por estos parámetros, lo que se muestra por medio de los puntos señalados con un '+' indicando que están encendidos y que corresponden al relleno del *wiggle*. Un valor de *fill* igual a -1, tendría el efecto de haber extendido el área de relleno de la traza afectando desde el punto 7 hasta el punto 12.

Por último, hay que hacer notar que dos o más trazas consecutivas pueden solaparse unas con otras en la gráfica final de salida. De esta manera, la apariencia de la sección sísmica tendría el siguiente aspecto:





## 4.2.3 Timming line.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Código del comando Timming line	2
2	Word Count	Número de palabras que siguen al comando	
3	Mask Number	Número de <i>timming line</i>	0-3
4	Y Start	Inicio de la máscara en Y	0-65535
5	X Start	Línea de inicio del <i>scan</i>	
6	X End	Línea de final del <i>scan</i>	
7	-----	No utilizado	
8	-----	No utilizado	
9	Time 1	Primera palabra del patrón de <i>timming</i>	
10	Time 2	Segunda palabra del patrón de <i>timming</i>	
	:		
	Time <i>n</i>	Enésima palabra del patrón de <i>timming</i>	

Este comando se utiliza ocasionalmente en el graficado de trazas sísmicas para sobreponer (overlay) trazas consecutivas. Al momento de procesar este comando, el sistema HSR-11B trabaja con grupos de 4 líneas de *scan* en la gráfica de salida, en cada una de las cuales se ubica el contorno de una traza, para sobreponerlas entre sí. Dentro del sistema de conversión este comando no se contempla debido a que las gráficas originales generadas en CAPS definen un espaciado uniforme de las trazas, y por lo tanto este comando nunca aparece en las secciones sísmicas que se procesan.

## 4.2.4 Serial data.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Modo de datos en serie	3
2	Word count	Número de palabras que siguen al comando	
3	Y Start	Punto de inicio en Y	
4	X Start	Línea de inicio del scan en X	
5	-----	No utilizado	
6	-----	No utilizado	
7	R Data 1	Palabra 1 de los datos tipo <i>raster</i>	
8	R Data 2	Palabra 2 de los datos tipo <i>raster</i>	
	:		
	R Data N	Palabra N de los datos tipo <i>raster</i>	

Este comando es similar al anterior, sólo que ahora la información gráfica de las trazas sísmicas que se desean combinar o sobreponer en la salida, son datos en serie provenientes del exterior a través de la interfaz del *Universal Graphic Bus* del HSR-11B. Al igual que en el caso anterior, este comando tampoco se procesa en el sistema de conversión debido a que toda la información que define la gráfica se encuentra en forma de archivo en disco.

### 4.2.5 Seismic flush.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Código del comando Seismic flush	4
2	Word count	Número de palabras que siguen al comando	2
3	Scans	Número de líneas de scan que se envían al puerto de salida.	

Este comando es utilizado junto con el modo de trazas sísmicas para dar salida hacia un dispositivo gráfico (por ejemplo un graficador), las líneas de *scan* que se encuentran almacenadas en un *buffer* interno del HSR-11B. Cada línea contiene el contorno de una traza sísmica. Una vez que este comando ha sido ejecutado, las memorias internas del rasterizador son limpiadas junto con varios registros como el Registro del Inicio de Línea, el cual es puesto a cero y así dejar listo el sistema para procesar una nueva gráfica.

Este comando no se considera en el programa de conversión, ya que en este caso la salida no se dirige hacia un dispositivo periférico en particular sino que se vuelve a escribir a disco pero en forma de archivo en formato CGM. Para el proceso de las trazas sísmicas sólo se considera el comando Seismic Traces y sus datos asociados (ver página 34).

#### 4.2.6 Graphic flush.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Código del comando Graphic flush	6
2	Word count	Número de palabras que siguen al comando	2
3	Scans	Número de líneas a vaciar	0-65535

Este comando es utilizado al final de la corriente de datos que define la gráfica para "limpiar" el remanente de líneas que pudieran quedar en la memoria interna del rasterizador. Una vez que este comando ha sido ejecutado todas las áreas de memoria son borradas y el registro interno que define el inicio de la siguiente línea de *scan* es puesto a cero.

Para el caso del programa de conversión, este comando es utilizado como una señal del fin del flujo de datos que definen la gráfica, y dependiendo del formato de archivo CGM que se desee generar, el programa se encarga de codificar en la salida la extensión final que tiene la gráfica a lo largo de los ejes, así como los delimitadores END PICTURE y END METAFILE (ver más adelante el capítulo 5) que marcan el final del metarchivo.

## 4.2.7 Restore raster buffer.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Modo de restablecimiento del buffer de la imagen rasterizada	5
2	Word count	Número de palabras que siguen al comando	
3	Y Start	Punto de inicio en el eje Y	
4	X Start	Número de la línea de scan	
5		Reservado	
6		Reservado	
7	Scan word 1	Primera palabra del scan	
8	Scan word 2	Segunda palabra del scan	
	:		
N	Scan word N	Última palabra de la línea de scan.	

Este comando se utiliza para restablecer la memoria intermedia (buffer) del rasterizador, la cual se utiliza de modo temporal para enviar los datos hacia un dispositivo gráfico de salida. Debido a que el sistema de conversión no produce salida hacia ningún dispositivo periférico sino únicamente a disco, este comando nunca aparece en los archivos que se procesan y por lo tanto no se contempla dentro del programa.

## 4.2.8 Sectored vectors.

Palabra (word)	Nombre	Significado	Rango de valores
1	Line thickness and command type	Byte alto: ancho de línea Byte bajo: código del comando Sectored vectors	0-7  7
2	Word count	Número de palabras que siguen al comando	
3	-----	No utilizado	
4	-----	No utilizado	
5	XS1	Valor de inicio en X del vector 1	0-65535
6	YS1	Valor de inicio en Y del vector 1	0-65535
7	XE1	Valor final en X del vector 1	0-65535
8	YE1	Valor final en Y del vector 1	0-65535
9	XS2	Valor de inicio en X del vector 2	0-65535
10	YS2	Valor de inicio en Y del vector 2	0-65535
11	XE2	Valor final en X del vector 2	0-65535
12	YE2	Valor final en Y del vector 2	0-65535
:			
:			
Etc.			

Este comando permite al usuario especificar un conjunto de vectores dadas sus coordenadas de inicio P1(XSn, YSn) y de final P2(XEn, YEn). El parámetro *line thickness* sirve para definir el ancho de línea con el cual se dibujarán los vectores, un valor de 0 ó 1 implica el utilizar un ancho de línea de un pixel.

Se deberá tener especial cuidado en que toda una serie de vectores quede dentro de los límites de la imagen, cualquier vector o grupo de vectores cuyas coordenadas se encuentren fuera del área de dibujo definido por el HSR-11B no serán trazados, siendo recortados en uno de sus bordes quedando el resto fuera de la gráfica. Por otra parte, un vector con un ancho de línea mayor a un pixel y una de sus coordenadas cercana al borde de la imagen, podría solaparse con el mismo, ocasionando lo que se conoce como *envolvimiento* (wrap-around), es decir, el ancho del vector que no alcanza a dibujarse fuera del borde, aparece dibujado en el lado opuesto de la gráfica.

#### 4.2.9 Scan line advance.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Código del comando Scan line advance	8
2	Word count	Número de palabras que siguen al comando	3
3	X Scan	Líneas de scan a avanzar en la salida	$N > 0$
4	Width	Ancho del bloque que se va avanzar en líneas de scan en memoria	Menor que el número de líneas de scan

Este comando solicita al sistema rasterizador que avance un determinado número de líneas hasta aquella cuyo número corresponda al definido para la variable X Scan. Si este valor es menor que el de la línea disponible de salida en ese momento, entonces el comando será ignorado. Este comando nunca aparece en las gráficas generadas en CAPS y por lo tanto el programa de conversión no lo procesa.



## 4.2.10 Initialize.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Código del comando Initialize	9
2	Word Count	Número de palabras que siguen al comando	
3	Scan start		
4	Output direction	Dirección de salida de las líneas de scan 0: de izquierda a derecha 1: de derecha a izquierda	0 ó 1
5	Scan length	Ancho de las líneas de scan	
6	Memory configuration	Organización de la memoria interna del rasterizador para procesar la imagen	0 - 7
7	Seismic direction	Incremento o decremento del eje Y	0 ó 1

Este comando siempre se encuentra al inicio del flujo de datos que definen la gráfica con el objeto de inicializar algunas variables de entorno como por ejemplo la dirección de salida de las líneas de *scan* que componen la gráfica o la organización interna de memoria del rasterizador que será utilizada para procesar la imagen. Sin embargo, el sistema de conversión no hace uso de ninguno de los parámetros definidos en este comando, en su lugar, el programa codifica en formato CGM un encabezado que incluye entre otras cosas, la precisión en números reales que se va a utilizar, la extensión en unidades de VDC, la tabla de colores a utilizar, junto con los elementos delimitadores BEGIN METAFILE, BEGIN PICTURE y BEGIN PICTURE BODY (ver más adelante en el capítulo 5 una explicación más detallada de estos elementos).

4.2.11 Alphanumerics.

Palabra (word)	Nombre	Significado	Rango de valores
1	Line thickness and command type	Byte alto: ancho de línea Byte bajo: código del comando Alphanumerics	0-15 10
2	Word count	Número de palabras que siguen al comando	N = (número de caracteres) / 15
3	Scale	Valor del tamaño de escala básico para el caracter	
4	Angle	Angulo de rotación	(0, 2, 4, 6)
5	X Start	Punto de inicio en X para la esquina inferior izquierda del primer caracter	
6	Y Start	Punto de inicio en Y para la esquia inferior izquierda del primer caracter	
7	-----	No utilizado	
8	-----	No utilizado	
9	C1, C2	1.- y 2.- caracteres ASCII	
10	C3, C4	3.- y 4.- caracteres ASCII	
	:		
N - 1		Últimos 2 caracteres	

Este comando hace que el sistema HSR-11B genere, escale y rote una cadena de caracteres ASCII, posicionando la esquina inferior izquierda de su primer carácter en el punto de coordenadas P(X Start, Y Start).

Un ángulo de rotación de 0 hace que los caracteres permanezcan en su posición por defecto a lo largo del eje X, en cambio un valor de 2 provoca un giro de 90° en el sentido de las manecillas del reloj, es decir, los caracteres se alinean ahora a lo largo del eje Y.

Se deberá tener especial cuidado en que todos los caracteres que componen una cadena sean posicionados y rotados de tal manera que todos queden dentro de los límites de la imagen, de otro modo se podría presentar un efecto de *envolvimiento* (wrap-around), dando como resultado que los segmentos de línea individuales que componen una imagen sean revertidos en su dirección y dispuestos en la coordenada errónea. Los límites que se deben manejar son los siguientes:

Dirección X	0 a 65535
Dirección Y	0 al número de bits por scan (ver comando número 9)

El parámetro *Line Thickness* especifica el ancho de la línea que será utilizado para dibujar los caracteres. Un valor de 0 ó 1 da como resultado que estos sean dibujados con un ancho de línea de un punto.

La representación interna de letras y números se da en una matriz de 9 x 9 puntos (ver la siguiente figura) con un espacio de dos columnas hacia la derecha para separar caracteres adyacentes. Un factor de escala de 1 hace referencia a una matriz de 9 x 9 para el dibujo de estos, en cambio un factor de escala de 2 daría caracteres en una matriz de 18 x 18, etc.

```

* * * * *
      *
. . . * . . .
. . . * . . .
. . . * . . .
. . . * . . .
. . . * . . .
. . . * . . .
. . . * . . .
. . . * . . .

```

El número disponible de bits por cada línea de scan se reducirá por 16 \* (número de coeficientes) si el comando *Seismic Data Resampling* es utilizado (ver más adelante una descripción de este comando).

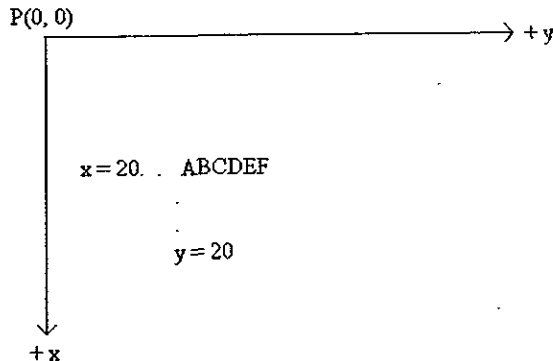
Cualquier carácter o partes del mismo que no puedan ser guardados dentro de las líneas de scan de la memoria interna del HSR-11B, serán recortadas y no se dibujarán en esa sección de la gráfica. El mismo comando Alphanumerics puede repetirse para otras secciones de la gráfica de manera tal que no importa que una cadena quede dentro de varios buffers, en la gráfica final de salida esta se dibujará correctamente.

Ejemplo del comando Alphanumerics.

Palabra (Word)	Value	Significado
1	10	Comando Alphanumerics
2	10	Word count (número de palabras que siguen al comando).
3	2	Scale Factor (factor de escala) = 2, matriz de 18 x 18
4	2	Angle (ángulo) = 2, rotación de 90
5	120	X Start = 120
6	20	Y Start = 20
7	----	No utilizado
8	----	No utilizado
9	AB	{ Cadena de
10	CD	{ caracteres ASCII
11	EF	{ "ABCDEF"

El resultado se observa en la siguiente figura:

El avance del papel se supone hacia arriba



Todos los caracteres se dibujan en una matriz de 18 x 18 puntos paralelas al eje y.

## 4.2.12 Resampling.

Palabra (word)	Nombre	Significado	Rango de valores
1	Command type	Código del comando Resampling	9
2	Word Count	Número de palabras que siguen al comando	
3	Points number	Número de puntos a generar	0-64
4	Coefficient number	Número de coeficientes de remuestreo usados por cada punto	6
5	-----	No utilizado	
6	-----	No utilizado	
7	S1N1	Coefficiente de remuestreo conjunto 1, número 1	
8	S1N2	Coefficiente de remuestreo conjunto 1, número 2	
9	S1N3	Coefficiente de remuestreo conjunto 1, número 3	
10	S1N4	Coefficiente de remuestreo conjunto 1, número 4	
11	S1N5	Coefficiente de remuestreo conjunto 1, número 5	
12	S1N6	Coefficiente de remuestreo conjunto 1, número 6	
13	S2N1	Coefficiente de remuestreo conjunto 2, número 1	
14	S2N2	Coefficiente de remuestreo. Conjunto 2, número 2	

---

Palabra (word)	Nombre	Significado	Rango de valores
15	S2N3	Coefficiente de remuestreo conjunto 2, número 3	
16	S2N4	Coefficiente de remuestreo conjunto 2, número 4	
17	S2N5	Coefficiente de remuestreo conjunto 2, número 5	
18	S2N6	Coefficiente de remuestreo conjunto 2, número 6	
	:		
	Etc.		

El remuestreo (resampling) sobre la información sísmica es una operación que se efectúa para hacer visibles al analista trazas que pudieran quedar oscurecidas debido a interferencias de ruidos o de otro tipo de señales. Este comando no se procesa por parte del programa de conversión debido a que los datos sísmicos que vienen en el archivo de entrada son manejados sin hacer ningún tipo de operación sobre los mismos.

---

## Capítulo 5

### Formato gráfico CGM (Computer Graphics Metafile)

El Metaarchivo para Gráficos por Computadora (CGM: Computer Graphics Metafile), es un formato gráfico definido en el estándar 8632 de la ISO (International Standards Organization) para la descripción, almacenamiento, recuperación e intercambio de información gráfica vectorizada, independiente de cualquier dispositivo de hardware. Para lograr lo anterior, dicho estándar define la forma (sintaxis) y el comportamiento funcional (semántica) de un conjunto de primitivas de graficación que pueden estar presentes en cualquier archivo gráfico apegado a este formato.

Toda imagen CGM es una secuencia de *elementos gráficos* los cuales están perfectamente diferenciados entre si de acuerdo a la *clase* a la que pertenecen y al *código de elemento* asociado a los mismos. Un solo metaarchivo puede contener varios cuadros o imágenes (frames) independientes entre si, o bien, como parte de una secuencia de animación con cada cuadro completamente separado uno del otro.

Debido a la característica vectorizada de este formato, en muchos casos existen múltiples maneras de especificar un mismo elemento, por ejemplo, un arco circular puede ser definido a partir de su centro, radio y un conjunto de líneas que cruzan por un punto inicial, intermedio y final, o bien, por medio de tres puntos por los cuales el arco debe pasar y utilizando un método de aproximación de curvas.

Todos los elementos gráficos presentes en una imagen CGM se construyen con base en un Dispositivo Virtual de Coordenadas (VDC: Virtual Device Coordinates), el cual constituye el área de trabajo en dos dimensiones donde se dibuja la imagen. El VDC definido para una gráfica puede tener cualquier extensión a lo largo de los ejes X e Y, independientemente de las capacidades de resolución del dispositivo físico de salida (pantalla o graficador). La extensión total del VDC para una imagen en particular es definida al momento de generarse y puede ser modificada posteriormente por los programas de visualización.

### 5.1 Estructura del metaarchivo.

Visto de un modo general, esto se ilustra en la figura 5.1. Las partes que aparecen escritas con letras mayúsculas corresponden a los llamados *elementos delimitadores* (o simplemente *delimitadores*) y que son estructuras que marcan los límites entre las partes principales que componen un metaarchivo. Estos se describirán con más detalle en las siguientes secciones de este mismo capítulo.

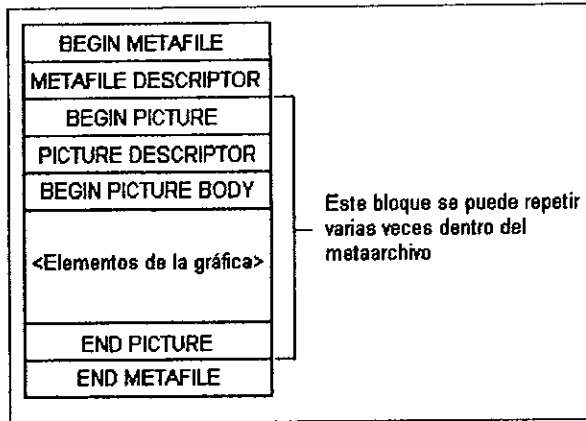


Figura 5.1. Estructura del metaarchivo CGM.

Aquí se observa que la parte correspondiente a la imagen gráfica se encuentra enmarcada por los elementos delimitadores BEGIN PICTURE y END PICTURE. Cada una de estas imágenes son independientes entre si, pudiendo existir varias de ellas almacenadas en un mismo metaarchivo. Al final, la aparición del delimitador END METAFILE señala el término de toda la secuencia de imágenes.

En la figura 5.2 se muestra un diagrama de estados que describe la estructura de los archivos CGM haciendo referencia a los distintos elementos que los integran, parte de ellos se describirán con más detalle en la siguiente sección, en especial aquellos que son utilizados por el programa de conversión en la construcción de las gráficas de secciones sísmicas.



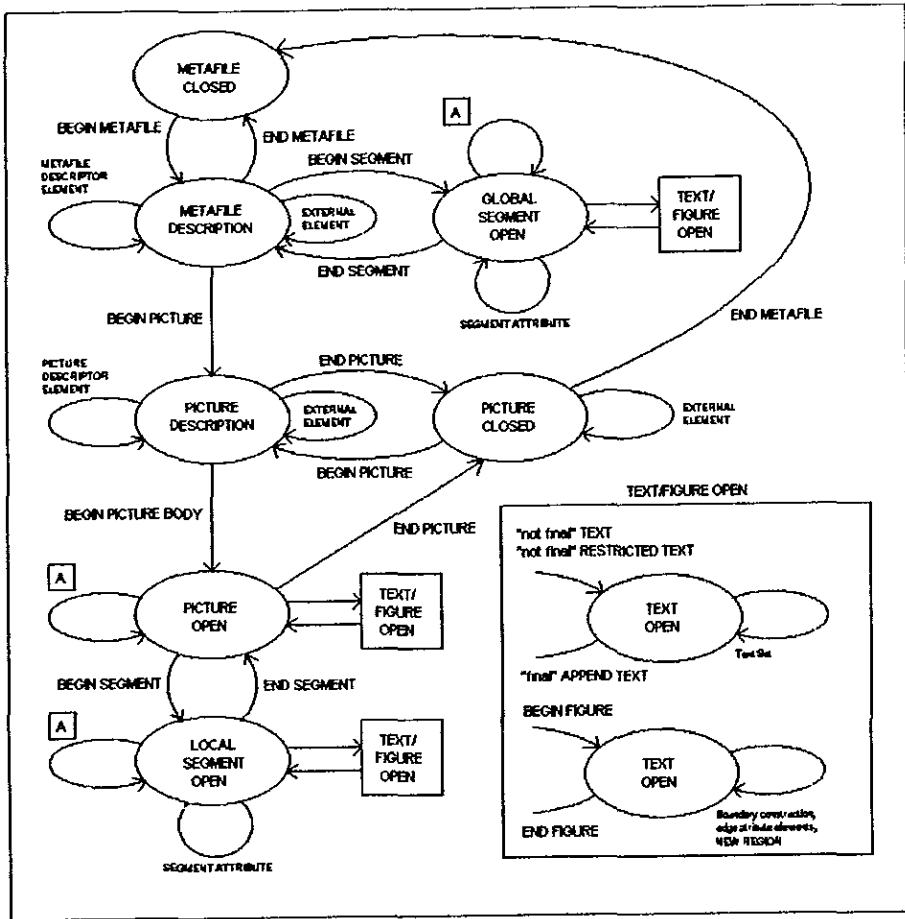


Figura 5.2. En este diagrama de estados se ilustra la secuencia de ocurrencia de los elementos que integran una imagen CGM. La letra A encerrada en un cuadrado incluye todos los elementos de control, primitivas gráficas, atributos, elementos externos y de escape, salvo *NEW REGION*, *CONNECTION EDGE*, *APPEND TEXT*, *"not final" TEXT*, *"not final" RESTRICTED TEXT*, *COPY SEGMENT*, *INHERITANCE FILTER* y *CLIP CONTROL*. El elemento *Text Set* incluye *"not final" APPEND TEXT*, *TEXT BUNDLE INDEX*, *TEXT FONT INDEX*, *TEXT PRECISION*, *CHARACTER EXPANSION FACTOR*, *CHARACTER SPACING*, *TEXT COLOUR*, *CHARACTER HEIGHT*, *CHARACTER SET INDEX*, *ALTERNATE CHARACTER SET INDEX*, *AUXILIARY COLOUR* y *TRANSPARENCY*.

El anterior diagrama fue tomado del documento oficial de referencia para el estándar CGM: American National Standards Institute/International Standards Organization, ANSI X3.122/ISO 8632. Information Processing Systems -Computer Graphics. Metafile for the storage and transfer of picture information. Part 1: Functional description. New York. p.p. 107. Ver bibliografía al final de la tesis.

## 5.2 Clases de elementos.

Todos los elementos gráficos presentes en un metaarchivo se encuentran agrupados en ocho diferentes clases, a cada uno de ellos se les asigna un código de elemento único para diferenciarlos entre sí, dentro de la clase a la que pertenecen. Por el momento las ocho clases definidas en el estándar son las siguientes:

Clase	Tipo de elementos
0	Elementos delimitadores (Delimiter Elements).
1	Descripción de elementos del metaarchivo (Metafile Descriptor Elements).
2	Descripción de elementos gráficos (Picture Descriptor Elements).
3	Elementos de control (Control Elements).
4	Elementos primitivos de la gráfica (Graphical Primitive Elements).
5	Atributos de los elementos (Attribute Elements).
6	Elementos de escape (Escape Elements).
7	Elementos externos (External Elements).
8-15	Reservados para futuras modificaciones.

El formato CGM define en total 158 elementos gráficos agrupados en las siete primeras clases de la tabla anterior. Hay que señalar que no todas las clases tienen el mismo número de elementos y una descripción completa para cada uno de ellos llevaría cientos de hojas en su descripción. Debido a que los archivos CGM que se generarán con el sistema de conversión sólo harán uso de un subconjunto de todos los elementos definidos en el estándar (20 en total), en los siguientes incisos se dará una lista de todos los elementos de cada clase, junto con una breve descripción de únicamente aquellos que se utilizarán en la generación de los archivos gráficos de secciones sísmicas.

### 5.2.1 Clase 0: elementos delimitadores.

Código del elemento	Nombre del elemento
1	BEGIN METAFILE
2	END METAFILE
3	BEGIN PICTURE
4	BEGIN PICTURE BODY
5	END PICTURE
6	BEGIN SEGMENT
7	END SEGMENT

**Elementos delimitadores (continuación).**

Código del elemento	Nombre del elemento
8	BEGIN FIGURE
9	END FIGURE
10	BEGIN PROTECTION REGION
11	END PROTECTION REGION
12	BEGIN COMPOUND LINE
13	END COMPOUND LINE
14	BEGIN COMPOUND TEXT PATH
15	END COMPOUND TEXT PATH
16	BEGIN TILE ARRAY
17	END TILE ARRAY

**BEGIN METAFILE**

Parámetros: cadena de caracteres.

Descripción: Este es el primer elemento presente en un metaarchivo CGM y deberá aparecer una sola vez dentro del mismo. El parámetro que le sigue depende de la modalidad de codificación que se esté utilizando y tiene el propósito de dar una breve descripción de su contenido.

**END METAFILE**

Parámetros: ninguno.

Descripción: Este es siempre el último elemento presente en un metaarchivo, y al igual que el anterior, deberá ocurrir sólo una vez dentro del mismo.

**BEGIN PICTURE**

Parámetros: cadena de caracteres.

Descripción: Este es el primer elemento de una imagen individual dentro de un metaarchivo CGM. Marca asimismo el inicio del bloque descriptor de la imagen (Picture Descriptor), y ocasiona que todos los elementos presentes en el mismo, incluidos los de control y de atributos, vuelvan a sus valores por defecto en caso de que hayan sido previamente redefinidos. La cadena de caracteres que se le pasa, al igual que con el elemento BEGIN METAFILE, no está aún estandarizada y sólo sirve para propósitos de esclarecimiento de la imagen almacenada.

**BEGIN PICTURE BODY**

Parámetros: ninguno.

Descripción: Este elemento señala el fin del bloque descriptor del *frame* (Picture Descriptor) y el inicio del cuerpo principal de la imagen. Asimismo, informa al interprete CGM de la transición del bloque descriptor de imagen a el bloque que contiene los las primitivas gráficas, los atributos y los elementos de control que definen una gráfica.

**END PICTURE**

Parámetros: ninguno.

Descripción: Representa el último elemento presente en un metaarchivo. Sólo elementos externos y de escape pueden aparecer entre los delimitadores END PICTURE y BEGIN PICTURE o entre los elementos END PICTURE y END METAFILE. La aparición de este elemento provoca un cambio de estado, según se ilustra en el diagrama 5.2, hacia la condición PICTURE CLOSED.

**5.2.2 Clase 1: descripción de elementos del metaarchivo.**

Código del elemento	Nombre del elemento
1	METAFILE VERSION
2	METAFILE DESCRIPTION
3	VDC TYPE
4	INTEGER PRECISION
5	REAL PRECISION
6	INDEX PRECISION
7	COLOUR PRECISION
8	COLOUR INDEX PRECISION
9	MAXIMUM COLOUR INDEX
10	COLOUR VALUE EXTENT
11	METAFILE ELEMENT LIST
12	METAFILE DEFAULTS REPLACEMENT
13	FONT LIST
14	CHARACTER SET LIST
15	CHARACTER CODING ANNOUNCER
16	NAME PRECISION
17	MAXIMUM VDC EXTENT
18	SEGMENT PRIORITY EXTENT
19	COLOUR MODEL
20	COLOUR CALIBRATION
21	FONT PROPERTIES

## Descripción de elementos del metaarchivo (continuación).

Código del elemento	Nombre del elemento
22	GLYPH MAPPING
23	SYMBOL LIBRARY LIST

### VDC TYPE

Parámetros: *integer* o *real*.

Descripción: el único parámetro que se le pasa a este elemento declara el tipo de datos que se utilizará en el Dispositivo Virtual de Coordenadas (VDC), el cual es el área de trabajo bidimensional en el que se dibujará la gráfica.

### REAL PRECISION

Parámetros: la forma del parámetro depende del tipo de codificación que se este usando.

Descripción: Este elemento establece la precisión y el rango de los operandos en números reales cuyos operadores empleen datos de este tipo. La forma del parámetro que se le pasa a este elemento es idéntica a la que se utiliza posteriormente para el elemento VDC REAL PRECISION. Esta es definida mediante tres campos numéricos, los dos primeros son reales y se refieren al rango de valores que se va a manejar, de menor a mayor. El tercero es un número entero que define la precisión, es decir, el número de dígitos decimales que se manejará como máximo

### METAFILE DEFAULTS REPLACEMENT

Parámetros: elementos descriptores de imagen, de control y de atributos.

Descripción: Este elemento encierra a su vez dentro de los delimitadores BEGMFDEFAULTS y ENDMFDEFAULTS una lista de elementos a los cuales se les asignan ciertos valores por defecto. Cada uno de ellos deberá conservar el mismo formato, significado y tener el mismo tipo de datos cuando aparezcan fuera de la definición dada por los anteriores delimitadores.

Todos los elementos en la lista son procesados secuencialmente; los parámetros que se le asignan a un elemento dependen de la posición de este último dentro de la lista. Así, si un mismo elemento aparece repetido entonces la última referencia que se tenga del mismo es la que se aplica en la descripción del metaarchivo.

### 5.2.3 Clase 2: descripción de elementos gráficos.

Código del elemento	Nombre del elemento
1	SCALING MODE
2	COLOUR SELECTION MODE
3	LINE WIDTH SPECIFICATION MODE
4	MARKER SIZE SPECIFICATION MODE
5	EDGE WIDTH SPECIFICATION MODE
6	VDC EXTENT
7	BACKGROUND COLOUR
8	DEVICE VIEWPORT
9	DEVICE VIEWPORT SPECIFICATION MODE
10	DEVICE VIEWPORT MAPPING
11	LINE REPRESENTATION
12	MARKER REPRESENTATION
13	TEXT REPRESENTATION
14	FILL REPRESENTATION
15	EDGE REPRESENTATION
16	INTERIOR STYLE SPECIFICATION MODE
17	LINE AND EDGE TYPE DEFINITION
18	HATCH STYLE DEFINITION
19	GEOMETRIC PATTERN DEFINITION

#### SCALING MODE

Parámetros: modo de escala métrica (*metric*) o abstracta (*abstract*). En caso de especificar escala métrica se incluye además un número real que define el factor de escala.

Descripción: El parámetro que acompaña a este elemento define el significado del VDC. Si se selecciona el modo '*abstract*', el espacio del VDC carece de dimensión y la imagen es correctamente dibujada a cualquier tamaño; dependiendo del dispositivo físico de salida (pantalla o graficador). En caso de seleccionarse el parámetro '*metric*', el espacio del VDC tiene medidas implícitas y representa distancia en milímetros. Una unidad de VDC equivale a un milímetro multiplicado por el factor de escala métrica, con lo cual la imagen es visualizada sólo a la escala indicada por este parámetro. Este elemento deberá aparecer en el bloque Descriptor de Imagen (Picture Descriptor) después del elemento delimitador BEGIN PICTURE y antes del delimitador BEGIN PICTURE BODY.

## VDC EXTENT

**Parámetros:** coordenadas de dos puntos que definen el área de trabajo donde se dibuja la imagen.  
**Descripción:** Estas dos coordenadas definen una extensión rectangular en el espacio del VDC el cual constituye la "región de interés" para los subsiguientes elementos del CGM. La primera coordenada representa, desde el punto de vista del observador, la esquina inferior izquierda de la imagen, la segunda la esquina superior derecha. Los valores de las coordenadas para cualquier dimensión pueden ambos incrementarse o disminuirse de la primera a la segunda coordenada. Por ejemplo, para dispositivos con un origen situado en la esquina superior izquierda, una imagen puede ser descrita en coordenadas que correspondan directamente a las del dispositivo aunque también podría ser visualizada correctamente en uno con origen en la esquina inferior izquierda.

De esta manera el VDC EXTENT establece un sentido de orientación para la dirección positiva de los ejes. Para el caso de un origen situado en la esquina superior izquierda, la dirección positiva de los ejes X e Y es hacia la derecha y hacia abajo respectivamente. Para un origen situado en la esquina inferior izquierda, las direcciones positivas son hacia la derecha para el eje X y hacia arriba para el eje Y, esta última disposición es la que más se utiliza en la mayoría de las imágenes CGM.

La especificación de valores para los elementos del CGM que queden fuera del rango marcado por el VDC EXTENT esta permitida. VDC EXTENT define únicamente la porción visible de la imagen, contenida dentro de los límites marcados por este elemento. Los valores numéricos X e Y deberán ser números enteros o reales dependiendo del parámetro especificado en el elemento VDC TYPE, y además dichos valores deberán estar dentro del rango definido por el elemento VDC REAL PRECISION.

## BACKGROUND COLOR

**Parámetros:** valor numérico del color.

**Descripción:** Este elemento y el valor que lo acompaña define el color de fondo de la imagen cuya definición inicia con el siguiente elemento BEGIN PICTURE BODY. El número que se le pasa a este elemento representa un valor directo de color, el cual es obtenido de una tabla de colores que se define mediante otro elemento llamado COLOUR TABLE (descrito más adelante en la clase 5).

### 5.2.4 Clase 3: elementos de control.

Código del elemento	Nombre del elemento
1	VDC INTEGER PRECISION
2	VDC REAL PRECISION
3	AUXILIARY COLOUR
4	TRANSPARENCY
5	CLIP RECTANGLE
6	CLIP INDICATOR
7	LINE CLIPPING MODE
8	MARKER CLIPPING MODE
9	EDGE CLIPPING MODE
10	NEW REGION
11	SAVE PRIMITIVE CONTEXT
12	RESTORE PRIMITIVE CONTEXT
13	PROTECTION REGION INDICATOR
14	GENERALIZED TEXT PATH MODE
15	MITRE LIMIT
16	TRANSPARENT CELL COLOUR

#### **VDC REAL PRECISION.**

**Parámetros:** la forma del parámetro depende del tipo de codificación que se este usando

**Descripción:** El parámetro que se le pasa a este elemento es idéntico en su forma al que se utiliza para REAL PRECISION. Básicamente el propósito de esta instrucción es el de definir el grado de "granularidad" que se tendrá dentro del área de trabajo establecido por VDC EXTENT. Esto último es establecido por medio de la precisión del número real que se desee manejar, de ahí que el número de decimales establecido para este elemento sea, de preferencia, idéntico al que se utilice para REAL PRECISION.



### 5.2.5 Clase 4: elementos primitivos de la gráfica.

Código del elemento	Nombre del elemento
1	POLYLINE
2	DISJOINT POLYLINE
3	POLYMARKER
4	TEXT
5	RESTRICTED TEXT
6	APPEND TEXT
7	POLYGON
8	POLYGON SET
9	CELL ARRAY
10	GENERALIZED DRAWING PRIMITIVE
11	RECTANGLE
12	CIRCLE
13	CIRCULAR ARC 3 POINT
14	CIRCULAR ARC 3 POINT CLOSE
15	CIRCULAR ARC CENTRE
16	CIRCULAR ARC CENTRE CLOSE
17	ELLIPSE
18	ELLIPTICAL ARC
19	ELLIPTICAL ARC CLOSE
20	CIRCULAR ARC CENTRE REVERSED
21	CONNECTING EDGE
22	HYPERBOLIC ARC
23	PARABOLIC ARC
24	NON-UNIFORM B-SPLINE
25	NON-UNIFORM RATIONAL B-SPLINE
26	POLYBEZIER
27	SYMBOL
28	BITONAL TILE
29	TILE

#### **POLYLINE**

**Parámetros:** lista de puntos de la forma  $P_1(X, Y) P_2(X, Y) P_3(X, Y) \dots P_n(X, Y)$ .

**Descripción:** Una línea es dibujada desde el primer punto en la lista de parámetros al segundo punto, del segundo al tercero y así sucesivamente hasta el último punto. La apariencia de la figura es controlada por los atributos de línea que se hayan establecido.

## DISJOINT POLYLINE

**Parámetros:** lista de puntos de la forma P1(X,Y) P2(X,Y) P3(X,Y) ... Pn(X,Y).

**Descripción:** una línea es dibujada desde el punto inicial en la lista de parámetros al segundo punto, otra desde el tercer punto hasta el cuarto, otra desde el quinto hasta el sexto punto y así sucesivamente hasta el final de la lista, formando una serie de segmentos de líneas rectas separadas unas de otras. Al igual que en el elemento anterior, la apariencia de todas las líneas dibujadas es controlada por los atributos de línea que se hayan establecido. Este elemento permite una considerable compresión de datos para aquellas aplicaciones que generen patrones de líneas, paneles o figuras de relleno de polígonos.

## POLYGON

**Parámetros:** lista de puntos de la forma P1(X,Y) P2(X,Y) P3(X,Y) ... Pn(X,Y).

**Descripción:** los límites de una región en forma de polígono son definidos por medio de la interconexión de cada uno de sus vértices con su sucesor en la anterior lista de puntos. Dicha unión se hace por medio de líneas rectas incluyendo el último vértice con el primero. El polígono así creado puede ser simple, es decir, de bordes rectos o bien con un redondeo en sus conexiones. El interior del mismo está definido por el elemento INTERIOR STYLE.

### 5.2.6 Clase 5: atributos de los elementos.

Código del elemento	Nombre del elemento
1	LINE BUNDLE INDEX
2	LINE TYPE
3	LINE WIDTH
4	LINE COLOUR
5	MARKER BUNDLE INDEX
6	MARKER TYPE
7	MARKER SIZE
8	MARKER COLOUR
9	TEXT BUNDLE INDEX
10	TEXT FONT INDEX
11	TEXT PRECISION
12	CHARACTER EXPANSION FACTOR
13	CHARACTER SPACING
14	TEXT COLOUR
15	CHARACTER HEIGHT

**Atributos de los elementos (continuación).**

<b>Código del elemento</b>	<b>Nombre del elemento</b>
16	CHARACTER ORIENTATION
17	TEXT PATH
18	TEXT ALIGMENT
19	CHARACTER SET INDEX
20	ALTERNATE CHARACTER SET INDEX
21	FILL BUNDLE INDEX
22	INTERIOR STYLE
23	FILL COLOUR
24	HATCH INDEX
25	PATTERN INDEX
26	EDGE BUNDLE INDEX
27	EDGE TYPE
28	EDGE WIDTH
29	EDGE COLOUR
30	EDGE VISIBILITY
31	FILL REFERENCE POINT
32	PATTERN TABLE
33	PATTERN SIZE
34	COLOUR TABLE
35	ASPECT SOURCE FLAGS
36	PICK IDENTIFIER
37	LINE CAP
38	LINE JOIN
39	LINE TYPE CONTINUATION
40	LINE TYPE INITIAL OFFSET
41	TEXT SCORE TYPE
42	RESTRICTED TEXT TYPE
43	INTERPOLATED INTERIOR
44	EDGE CAP
45	EDGE JOIN
46	EDGE TYPE CONTINUATION
47	EDGE TYPE INITIAL OFFSET
48	SYBOL LIBRARY INDEX
49	SYMBOL COLOUR
50	SIMBOL SIZE
51	SIMBOL ORIENTATION

## LINE WIDTH

**Parámetros:** número que especifica el ancho de línea en unidades de VDC.

**Descripción:** este elemento define el grueso de las líneas que serán dibujadas en los trazos de la gráfica. Afecta principalmente a los elementos POLYLINE y DISJOINT POLYLINE.

## LINE COLOUR

**Parámetros:** número del color a utilizar referido a una tabla o directamente mediante sus componentes RGB.

**Descripción:** este elemento especifica el color que será utilizado para las líneas. Si previamente se ha definido una tabla de colores personalizados por el usuario, el número que se le pasa a este elemento hace referencia al índice dentro de dicha tabla. En caso contrario, se le pasa como parámetro una terna de valores que indican las componentes RGB del color que se desea. Cada componente deberá estar en el rango de 0 a 255.

## INTERIOR STYLE

**Parámetros:** estilo del patrón de relleno, puede ser *hollow*, *solid pattern*, *hatch*, *empty*, *geometric pattern* o *interpolated*.

**Descripción:** Este elemento define el estilo de relleno de acuerdo a un patrón previamente establecido. Algunos de los elementos que son afectados por esta instrucción son por ejemplo POLYGON, POLYGON SET, RECTANGLE, CIRCLE, CIRCULAR ARC CLOSE, ELLIPSE Y ELLIPTICAL ARC CLOSE.

## FILL COLOUR

**Parámetros:** número del color a utilizar referido a una tabla o directamente mediante sus componentes RGB.

**Descripción:** este elemento es semejante al de Line Colour, solo que en este caso el color se refiere al relleno de elementos tales como polígonos, círculos, elipses, rectángulos, etc

## COLOUR TABLE

**Parámetros:** índice de inicio seguido de la definición de los colores mediante ternas de valores RGB para cada uno de ellos.

**Descripción:** la tabla de colores inicia con un número que sirve como un índice de inicio para cada color definido mediante una terna de valores RGB, cada componente deberá estar en el rango de 0 a 255. El color 0 normalmente se asocia al color de fondo.

### 5.2.7 Clase 6: elementos de escape

Código del elemento	Nombre del elemento
1	ESCAPE

### 5.2.8 Clase 7: elementos externos.

Código del elemento	Nombre del elemento
1	MESSAGE
2	APPLICATION DATA

## 5.3 Formas de codificación.

El estándar CGM define tres diferentes formas de codificación para los archivos gráficos: binario, en modo de texto y en modo carácter. Este último es el más compacto en tamaño, estando orientado su uso para la transmisión de información gráfica por medio de dispositivos de tipo alfanumérico, ya que bajo esta modalidad todos los elementos gráficos son almacenados por medio de un código de caracteres de 7 bits definido por la ISO en su estándar 646.

La codificación binaria, por su parte, es la más comúnmente utilizada por las aplicaciones, ofrece un tamaño muy reducido en sus archivos de salida pero un poco más grandes que los generados con la codificación en modo carácter. La imagen gráfica se almacena como una secuencia de octetos (bytes) y de palabras (words) de 16 y 32 bits de longitud, lo cual facilita mucho la tarea de generar y decodificar metaarchivos almacenados en esta forma.

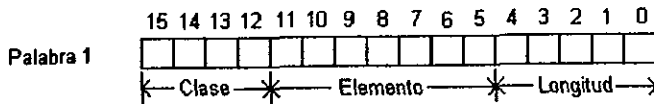
Por último, la codificación en modo de texto es una opción que se utiliza principalmente para propósitos de fácil modificación y lectura de los elementos que integran una gráfica, en este caso la imagen se almacena como un archivo de texto estándar —humanamente legible, que contiene una descripción explícita de todos los elementos gráficos que se encuentran codificados en el metaarchivo.

Ya que el sistema de conversión está orientado principalmente a la creación de archivos binarios CGM de las gráficas de secciones sísmicas, en la siguiente sección se describirá en detalle esta forma de codificación. En las secciones 5.3.3 y 5.3.4 se describe brevemente la codificación en modo de texto y en modo carácter, respectivamente. El lector interesado en los detalles de implementación de estas modalidades de codificación puede acudir a la fuente de referencia del estándar CGM incluida en la bibliografía al final de la tesis.

### 5.3.1 Metaarchivo binario.

La codificación binaria provee una sintaxis en su representación que puede ser fácilmente optimizada en tamaño, con el propósito de lograr una máxima velocidad en la generación e interpretación de las imágenes, al mismo tiempo que proporciona una codificación estándar intercambiable entre diferentes sistemas de cómputo.

Todos los elementos gráficos se representan por medio de *comandos*, los cuales pueden ser de forma corta (short form commands) o extendida (long form commands). Cada *comando*, independientemente de su forma, empieza con un encabezado (header) de 16 bits de longitud, dividido en tres campos según se muestra a continuación para el caso de los comandos en forma corta:

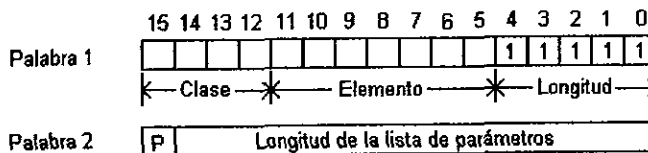


Forma corta de un encabezado de comando

Los campos que lo componen son los siguientes:

- Bits 15-12:** Clase del elemento (en un rango de valores de 0 a 15).
- Bits 11-5:** Código del elemento (en un rango de 0 a 127).
- Bits 4-0:** Longitud de la lista de parámetros, es decir, el número de bytes que siguen al encabezado y que constituyen los datos de entrada del elemento gráfico codificado (en un rango de valores de 0 a 30).

Por su parte, la codificación de un comando en su forma extendida tiene la siguiente estructura:



Formato extendido para un encabezado de comando

Los campos presentes en un comando en forma extendida son los siguientes:

#### Palabra 1

- Bits 15-12: Clase del elemento (en un rango de valores de 0 a 15).
- Bits 11-5: Código del elemento (en un rango de 0 a 127).
- Bits 4-0: Valor binario 11111 (decimal 31) que indica forma extendida.

#### Palabra 2

- Bit 15: Bandera de partición.
  - 0 para la última partición.
  - 1 para una partición intermedia.
- Bits 14-0: Longitud de la lista de parámetros, es decir, el número de bytes de datos asociados al elemento gráfico en cuestión y que siguen al encabezado, en un rango de 0 a 32767.

Como se puede observar, la primera palabra de un comando en forma extendida es idéntica en su estructura a su equivalente en forma corta. La presencia del valor binario 11111 (31 en decimal) en el campo longitud de la lista de parámetros, indica precisamente que se trata de un comando en forma extendida. Por lo tanto, el encabezado de comando en este caso consiste de dos palabras, donde la segunda contiene la verdadera longitud (en bytes) del número de elementos de datos que siguen al comando.

Adicionalmente a lo anterior, los comandos en forma extendida permiten también que la longitud de la lista de parámetros sea particionada. El bit 15 de la segunda palabra indica si la longitud dada por los 14 bits restantes completa el número de elementos de datos que siguen al comando, o bien, si continúan en otra palabra adicional. Para particiones subsecuentes, los 14 primeros bits de cada palabra se van añadiendo hacia la izquierda de la palabra anterior. La partición final de un elemento es indicada por el bit 15 cuando este es igual a cero.

### Tipos y tamaños de datos.

La codificación binaria del CGM utiliza además cinco tipos de datos básicos para describir los diferentes parámetros de un elemento gráfico, representados por los siguientes símbolos de acuerdo al estándar:

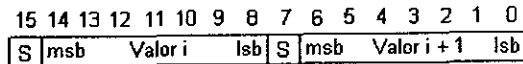
- SI    Números enteros con signo (Signed Integers).
- UI    Números enteros sin signo (Unsigned Integers).
- C    Caracter
- FX    Números reales de punto fijo (Fixed Point Real).
- FP    Números reales de punto flotante (Floating Point Real).

### Números enteros con signo.

Estos son representados en complemento a dos y en cuatro diferentes precisiones: 8, 16, 24 y 32 bits. Hay que señalar que los elementos de datos que requieran parámetros de este tipo no pueden usar la forma de 8 bits.

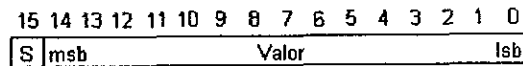
#### Valor entero con signo de 8 bits.

Ocupa media palabra de longitud (un byte).



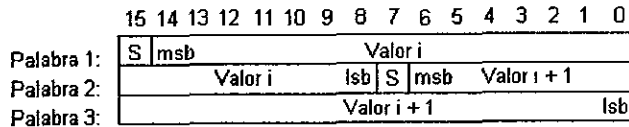
#### Valor entero con signo de 16 bits.

Abarca una palabra completa.



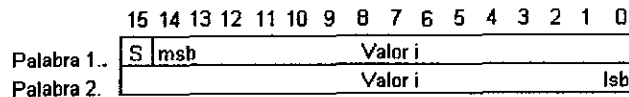
#### Valor entero con signo de 24 bits.

Cada número ocupa una palabra y media (3 bytes) de la siguiente manera.



#### Valor entero con signo de 32 bits.

Ocupa dos palabras completas del metaarchivo

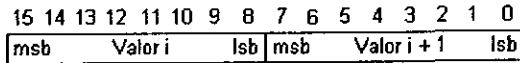




**Números enteros sin signo.**

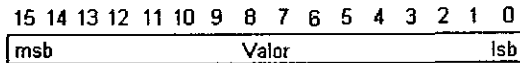
**Valor entero sin signo de 8 bits.**

Ocupa media palabra de longitud (un byte)



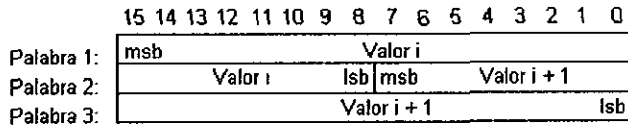
**Valor entero sin signo de 16 bits.**

Abarca una palabra completa



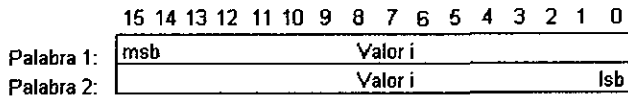
**Valor entero sin signo de 24 bits.**

Cada número ocupa una palabra y media (3 bytes) de la siguiente manera:



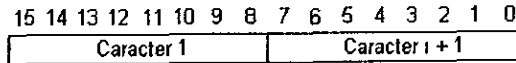
**Valor entero sin signo de 32 bits.**

Ocupa dos palabras completas del metaarchivo.



## Caracter

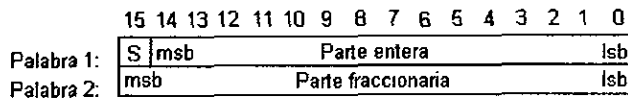
Cada caracter es almacenado en un solo byte:



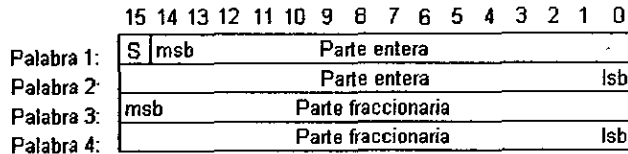
## Números reales de punto fijo.

Este tipo de números es almacenado mediante dos valores enteros, el primero representa la “parte entera” y tiene la misma forma que un número SI; el segundo representa la “parte fraccionaria” y tiene idéntica estructura a la de una valor entero sin signo (UI). Dos precisiones son permitidas para los números reales de punto fijo: 32 y 64 bits.

### Valor real de punto fijo de 32 bits.



### Valor real de punto fijo de 64 bits.



Los valores así representados quedan definidos mediante las siguientes expresiones.

Para 32 bits :  $\text{valor\_real} = SI + \left( \frac{UI}{2^{16}} \right)$

Para 64 bits :  $\text{valor\_real} = SI + \left( \frac{UI}{2^{32}} \right)$

Nota: la “parte entera” (representada por el valor SI) es el mayor entero menor que o igual al valor real que se pretende representar.

## Números reales de punto flotante.

Estos números se representan mediante el formato P754 definido por la IEEE (Institute of Electrical and Electronics Engineers) para valores reales de punto flotante. Este formato consiste de tres partes:

- Un bit de signo 's'.
- Una parte exponencial 'e'.
- Una parte fraccionaria 'f'.

El valor así obtenido es una función de estos tres elementos. Si 's' es 0, el valor es positivo; si es 1, el valor es negativo. Dos precisiones están disponibles para este tipo de números: 32 y 64 bits. La magnitud del valor es calculada como sigue para el caso de una representación de 32 bits:

- a) Si  $e = 255$  y  $f \neq 0$ , entonces el valor es indefinido.
- b) Si  $e = 255$  y  $f = 0$ , entonces el valor es el máximo número positivo ( $s = 0$ ) o negativo ( $s = 1$ ) permitido.
- c) Si  $0 < e < 255$ , entonces la magnitud del valor es  $(1.f)(2^{e-127})$ .
- d) Si  $e = 0$  y  $f \neq 0$ , entonces la magnitud del valor es de  $(0.f)(2^{e-126})$ .
- e) Si  $e = 0$  y  $f = 0$ , entonces el valor es 0.

La magnitud del valor para el caso de una representación de 64 bits es calculada de la siguiente manera:

- f) Si  $e = 2047$  y  $f \neq 0$ , entonces el valor es indefinido.
- g) Si  $e = 2047$  y  $f = 0$ , entonces el valor es el máximo número positivo ( $s = 0$ ) o negativo ( $s = 1$ ) permitido.
- h) Si  $0 < e < 2047$ , entonces la magnitud del valor es  $(1.f)(2^{e-1023})$ .
- i) Si  $e = 0$  y  $f \neq 0$ , entonces la magnitud del valor es de  $(0.f)(2^{e-1022})$ .
- j) Si  $e = 0$  y  $f = 0$ , entonces el valor es 0.

### Valores reales de punto flotante de 32 bits.

Ocupan dos palabras de 16 bits y se dividen en tres campos como sigue:

Signo	1 bit
Exponente	8 bits.
Fracción	23 bits.

Su representación interna es como sigue

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Palabra 1:	S	msb					Exponente					lsb	msb					Fracción					
Palabra 2:	Fracción															lsb							

**Valores reales de punto flotante de 64 bits.**

Ocupan en total cuatro palabras de 16 bits y también se dividen en tres campos de la siguiente manera:

- Signo            1 bit.
- Exponente    11 bits.
- Fracción      52 bits.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Palabra 1:	S	msb					Exponente					lsb	msb					F
Palabra 2:	Fracción																	
Palabra 3:	Fracción																	
Palabra 4:	Fracción															lsb		

**Ejemplos de codificación.**

Los siguientes ejemplos ilustran el uso de la codificación binaria del CGM para algunos de los elementos gráficos definidos en el estándar.

**Ejemplo 1. BEGIN METAFILE 'Example 1'**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Encabezado:	0					1					10					
Longitud:	9										'E'					
Nombre:	'x'										'a'					
	'm'										'p'					
	'l'										'e'					
	''										'l'					

Ejemplo 2 BEGIN PICTURE 'Test'

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Encabezado:	0				3				5							
Longitud:	4								T							
Nombre:	'e'								's'							
	Y								0							

Ejemplo 3. POLYLINE desde el punto P<sub>1</sub>(0,2) al punto P<sub>2</sub>(1,3) al punto P<sub>3</sub>(2,1) y finalmente al punto P<sub>4</sub>(0,2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Encabezado:	4				1				16							
Punto P <sub>1</sub> (0, 2):									0							
									2							
Punto P <sub>2</sub> (1, 3):									1							
									3							
Punto P <sub>3</sub> (2, 1):									2							
									1							
Punto P <sub>4</sub> (0, 2):									0							
									2							

Ejemplo 4. POLYLINE particionado con 50 puntos.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Encabezado:	4				1				31								
Long (cont.):	1									120							
Punto 1:									x(1)								
									y(1)								
Punto 30:									x(30)								
									y(30)								
Long (final):	0									80							
Punto 31:									x(31)								
									y(31)								
Punto 50:									x(50)								
									y(50)								

Ejemplo 5. METAFILE DEFAULTS REPLACEMENT linewidth 0.5

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Encabezado:	1			11						4						
Line width:	5			3						2						
Valor:	0.5															

5.3.2 Metaarchivo en modo de texto.

La codificación en modo de texto (Clear Text Encoding) provee una representación de las imágenes CGM fácil de leer, editar y codificar ya que permite que las gráficas sean creadas o modificadas a partir de cualquier editor de texto estándar, utilizando para ello el conjunto de caracteres interno del sistema anfitrión.

Cualquier metaarchivo CGM codificado en esta forma consiste de un flujo de caracteres que dan forma a una serie de elementos gráficos. Cada elemento gráfico se especifica mediante una abreviatura del mismo, seguido de sus parámetros y finalizando con un caracter delimitador que lo separa del siguiente elemento codificado. Dicho caracter puede ser la diagonal invertida ‘\’ o el punto y coma ‘;’.

La codificación de los tipos de datos básicos, en lo que se refiere a valores enteros y de punto flotante, es en forma directa y en notación decimal, tal y como se utiliza normalmente. Los arreglos de hileras (strings) se enmarcan por medio de apóstrofes ‘ ’ y el caracter separador utilizado entre elementos es el espacio en blanco.

Abreviaturas utilizadas

La siguiente tabla proporciona la equivalencia entre los elementos gráficos definidos en el estándar y su abreviatura, tal y como es utilizada en la codificación en modo de texto.

Nombre del elemento	Abreviatura
BEGIN METAFILE	BEGMF
END METAFILE	ENDMF
BEGIN PICTURE	BEGPIC
BEGIN PICTURE BODY	BEGPICBODY
END PICTURE	ENDPIC
METAFILE VERSION	MFVERSION
METAFILE DESCRIPTION	MFDESC
VDC TYPE	VDCTYPE

## Abreviaturas utilizadas (continuación).

Nombre del elemento	Abreviatura
INTEGER PRECISION	INTEGERPREC
REAL PRECISION	REALPREC
INDEX PRECISION	INDEXPREC
COLOUR PRECISION	COLRPC
COLOUR INDEX PRECISION	COLRINDEXPREC
MAXIMUM COLOUR INDEX	MAXCOLRINDEX
COLOUR VALUE EXTENT	COLRVALUEEXT
METAFILE ELEMENT LIST	MFELEMLIST
METAFILE DEFAULTS REPLACEMENT	BEGMFDEFAULTS ENDMFDEFAULTS
FONT LIST	FONTLIST
CHARACTER SET LIST	CHARSETLIST
CHARACTER CODING ANNOUNCER	CHARCODING
SCALING MODE	SCALEMODE
COLOUR SELECTION MODE	COLRMODE
LINE WIDTH SPECIFICATION MODE	LINEWIDTHMODE
MARKER SIZE SPECIFICATION MODE	MARKERSIZEMODE
EDGE WIDTH SPECIFICATION MODE	EDGEWIDTHMODE
VDC EXTENT	VDC EXT
BACKGROUND COLOUR	BACKCOLR
VDC INTEGER PRECISION	VDCINTEGERPREC
VDC REAL PRECISION	VDCREALPREC
AUXILIARY COLOUR	AUXCOLR
TRANSPARENCY	TRANSPARENCY
CLIP RECTANGLE	CLIPRECT
CLIP INDICATOR	CLIP
POLYLINE	LINE INCRLINE
DISJOINT POLYLINE	DISJTLINE INCRDISJTLINE
POLYMARKER	MARKER INCRMARKER
TEXT	TEXT
RESTRICTED TEXT	RESTRTEXT
APPEND TEXT	APNDTEXT
POLYGON	POLYGON INCRPOLYGON
POLYGON SET	POLYGONSET INCRPOLYGONSET

## Abreviaturas utilizadas (continuación).

Nombre del elemento	Abreviatura
CELL ARRAY	CELLARRAY
GENERALIZED DRAWING PRIMITIVE	GDP
RECTANGLE	RECT
CIRCLE	CIRCLE
CIRCULAR ARC 3 POINT	ARC3PT
CIRCULAR ARC 3 POINT CLOSE	ARC3PTCLOSE
CIRCULAR ARC CENTRE	ARCCTR
CIRCULAR ARC CENTRE CLOSE	ARCTRCLOSE
ELLIPSE	ELLIPSE
ELLIPTICAL ARC	ELLIPARC
ELLIPTICAL ARC CLOSE	ELLIPARCCLOSE
LINE BUNDLE INDEX	LINEINDEX
LINE TYPE	LINETYPE
LINE WIDTH	LINEWIDTH
LINE COLOUR	LINECOLR
MARKER BUNDLE INDEX	MARKERINDEX
MARKER TYPE	MARKERTYPE
MARKER SIZE	MARKERSIZE
MARKER COLOUR	MARKERCOLR
TEXT BUNDLE INDEX	TEXTINDEX
TEXT FONT INDEX	TEXTFONTINDEX
TEXT PRECISION	TEXTPREC
CHARACTER EXPANSION FACTOR	CHAREXPAN
CHARACTER SPACING	CHARSPACE
TEXT COLOUR	TEXTCOLR
CHARACTER HEIGHT	CHARHEIGHT
CHARACTER ORIENTATION	CHARORI
TEXT PATH	TEXTPATH
TEXT ALIGNMENT	TEXTALIGN
CHARACTER SET INDEX	CHARSETINDEX
ALTERNATE CHARACTER SET INDEX	ALTCHARSETINDEX
FILL BUNDLE INDEX	FILLINDEX
INTERIOR STYLE	INTSTYLE
FILL COLOUR	FILLCOLR
HATCH INDEX	HATCHINDEX
PATTERN INDEX	PATINDEX
EDGE BUNDLE INDEX	EDGEINDEX
EDGE TYPE	EDGETYPE
EDGE WIDTH	EDGEWIDTH
EDGE COLOUR	EDGECOLR

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**



**Abreviaturas utilizadas (continuación).**

Nombre del elemento	Abreviatura
EDGE VISIBILITY	EDGEVIS
FILL REFERENCE POINT	FILLREFPT
PATTERN TABLE	PATTABLE
PATTERN SIZE	PATSIZE
COLOUR TABLE	COLRTABLE
ASPECT SOURCE FLAGS	ASF
ESCAPE	ESCAPE
MESSAGE	MESSAGE
APPLICATION DATA	APPLDATA

**Ejemplo de codificación.**

El siguiente es un ejemplo que ilustra cómo se ve un metaarchivo CGM codificado en forma de texto. En la página 85 se observa la imagen gráfica que representa.

**Archivo HELIXSP.CGM**

```

BEGMF ";
mfversion 1;
mfdesc ";
mfefemlist 'DRAWINGSET BEGMFDEFAULTS ENDMFDEFAULTS SCALEMODE
INTEGERPREC
    REALPREC';
BEGMFDEFAULTS;
    vdcintegerprec -512, 511;
    vdcext (0,0) (4095,4095);
ENDMFDEFAULTS;
integerprec -16384, 16383;
realprec -67108864.00000000, 67108864.00000000, 8 % 26 binary bits %;

BEGPIC ";
vdcext (0,0) (884,1144);
scalemode metric, 0 24420023;
BEGPICBODY;
LINE (0,0) (0,1144) (884,1144) (884,0) (0,0),
DISJTLINE (178,442) (172,438) (166,430) (165,433);
LINE (169,437) (161,437) (162,433) (163,434);
LINE (168,444) (167,443) (166,444) (167,445) (168,444);

```

LINE (166,447) (167,448) (166,450) (165,450) (163,449) (162,448)  
(163,445) (160,443) (159,447);  
LINE (178,442) (168,494) (163,491) (168,494);  
DISJTLINE (157,476) (156,480);  
LINE (159,483) (160,485) (159,487) (158,487) (156,486) (155,484)  
(156,482) (152,480) (152,484);  
LINE (158,491) (157,490) (157,491) (158,492) (158,491),  
LINE (157,494) (151,492) (150,492) (150,494) (151,495) (156,498)  
(157,498) (157,496) (157,494);  
LINE (168,494) (158,549) (152,546) (158,549);  
DISJTLINE (146,531) (145,535);  
LINE (143,536) (141,537) (141,539) (142,540) (144,541) (148,539)  
(149,539) (148,543) (147,543),  
LINE (148,547) (147,546) (146,547) (148,548) (148,547),  
LINE (146,551) (147,552) (146,554) (145,555) (143,554) (142,552)  
(143,549) (139,547) (138,551);  
LINE (158,549) (147,609) (141,606) (147,609);  
LINE (137,598) (131,596) (130,596) (129,598) (130,600) (136,603)  
(137,602) (138,600) (137,598);  
LINE (136,607) (135,606) (135,607) (136,608) (136,607);  
LINE (134,611) (128,608) (127,609) (127,611) (128,613) (133,615)  
(135,615) (135,613) (134,611);  
LINE (147,609) (135,673) (129,670) (135,673),  
LINE (118,660) (117,660) (117,663) (117,664) (120,665) (124,662)  
(126,663) (125,667) (123,667);  
LINE (124,671) (123,671) (122,672) (124,673) (124,671);  
LINE (122,676) (123,678) (122,680) (121,681) (118,680) (117,678)  
(118,674) (115,673) (114,678),  
LINE (135,673) (122,742) (115,740) (122,742);  
LINE (111,731) (112,733) (112,735) (110,736) (108,735) (107,733)  
(107,730) (104,728) (103,733);  
LINE (111,741) (109,740) (109,742) (110,742) (111,741);  
LINE (108,746) (102,744) (100,744) (100,747) (101,749) (107,751)  
(109,750) (109,748) (108,746);  
LINE (122,742) (109,817) (101,814) (109,817);  
LINE (98,806) (88,808) (89,803) (90,803);  
LINE (96,816) (95,815) (95,817) (96,817) (96,816);  
LINE (94,822) (95,823) (94,826) (93,827) (90,826) (89,824) (90,820)  
(86,819) (85,824);  
LINE (109,817) (94,897) (86,895) (94,897),  
LINE (75,881) (73,882) (83,885);  
DISJTLINE (83,884) (83,887);  
LINE (80,890) (74,888) (72,889) (71,892) (72,894) (79,896) (81,895)  
(82,892) (80,890);  
LINE (80,902) (78,901) (78,903) (79,903) (80,902);

LINE (77,908) (70,906) (69,907) (68,910) (69,912) (76,914) (78,913)  
(78,910) (77,908);  
LINE (105,621) (103,620) (102,626) (115,625) (114,631) (112,631);  
DISJTLINE (468,188) (474,182);  
LINE (468,188) (526,239) (532,234) (526,239);  
LINE (538,221) (540,221) (542,223) (542,224) (540,226) (538,226)  
(535,224) (531,226) (535,230),  
LINE (547,227) (546,228) (547,229) (548,228) (547,227);  
LINE (550,232) (544,236) (544,238) (546,240) (548,240) (554,235)  
(554,233) (552,232) (550,232);  
LINE (526,239) (580,286) (586,281) (580,286),  
LINE (586,273) (586,275) (588,277) (590,277) (592,275) (592,269)  
(593,268) (597,272) (596,272);  
LINE (600,275) (599,275) (600,276) (601,275) (600,275);  
LINE (603,279) (605,279) (607,281) (607,282) (604,284) (602,284)  
(600,282) (596,284) (600,287);  
LINE (580,286) (630,329) (636,325) (630,329),  
LINE (642,314) (636,318) (636,319) (638,321) (640,321) (645,317)  
(645,315) (644,314) (642,314);  
LINE (649,319) (648,320) (649,320) (650,320) (649,319);  
LINE (652,323) (647,327) (646,328) (648,330) (650,330) (655,326)  
(656,325) (654,323) (652,323);  
LINE (630,329) (676,369) (682,365) (676,369);  
DISJTLINE (680,353) (683,356);  
LINE (682,359) (682,360) (684,362) (686,362) (688,360) (688,355)  
(689,354) (692,357) (691,358);  
LINE (695,360) (694,360) (695,361) (696,360) (695,360),  
LINE (697,364) (699,364) (701,365) (701,366) (698,368) (697,368)  
(694,366) (691,368) (694,371);  
LINE (676,369) (718,407) (724,403) (718,407);  
DISJTLINE (723,391) (726,394);  
LINE (731,393) (733,393) (734,395) (734,396) (732,397) (730,397)  
(728,395) (724,397) (727,400);  
LINE (737,398) (736,398) (737,399) (738,398) (737,398);  
LINE (740,401) (734,405) (734,406) (735,407) (737,407) (743,404)  
(743,403) (741,401) (740,401),  
LINE (718,407) (758,442) (764,438) (758,442);  
DISJTLINE (763,427) (766,430);  
LINE (772,429) (767,435) (764,433) (765,432);  
LINE (777,433) (776,434) (777,434) (778,434) (777,433),  
LINE (779,437) (781,437) (782,438) (782,439) (779,440) (778,440)  
(776,438) (773,440) (775,443);  
DISJTLINE (664,302) (657,314) (653,309) (668,306) (178,442) (172,438),  
LINE (178,442) (218,407) (212,403) (218,407),  
DISJTLINE (197,406) (200,403);

POLYGON (606,1031) (607,1033) (608,1035) (610,1036) (612,1037) (614,1038)  
(616,1038) (619,1038) (620,1037) (622,1036) (624,1034) (625,1033) (625,1031)  
(626,1029) (626,1027) (626,1025) (609,1025) (609,1024) (610,1023) (610,1022)  
(610,1022) (611,1020) (612,1019) (613,1019) (614,1018) (615,1018) (617,1018)  
(618,1018) (619,1019) (620,1020) (621,1021) (622,1022) (626,1022) (625,1020)  
(625,1019) (624,1018) (623,1017) (621,1015) (619,1014) (617,1014) (615,1014)  
(612,1015) (610,1016) (608,1017) (607,1018) (607,1019) (606,1021) (606,1022)  
(605,1024) (605,1027) (605,1029) (606,1031) (609,1028) (621,1028) (621,1029)  
(621,1030) (621,1030) (621,1031) (621,1032) (620,1033) (619,1033) (617,1034)  
(616,1034) (615,1034) (614,1034) (612,1033) (611,1033) (611,1032) (610,1031)  
(610,1030) (610,1030) (609,1029) (609,1028) (606,1031);

ESCAPE -802 ";

LINE (606,1031) (607,1033) (608,1035) (610,1036) (612,1037) (614,1038)  
(616,1038) (619,1038) (620,1037) (622,1036) (624,1034) (625,1033) (625,1031)  
(626,1029) (626,1027) (626,1025) (609,1025) (609,1024) (610,1023) (610,1022)  
(610,1022) (611,1020) (612,1019) (613,1019) (614,1018) (615,1018) (617,1018)  
(618,1018) (619,1019) (620,1020) (621,1021) (622,1022) (626,1022) (625,1020)  
(625,1019) (624,1018) (623,1017) (621,1015) (619,1014) (617,1014) (615,1014)  
(612,1015) (610,1016) (608,1017) (607,1018) (607,1019) (606,1021) (606,1022)  
(605,1024) (605,1027) (605,1029) (606,1031);

ESCAPE -802 ";

LINE (609,1028) (621,1028) (621,1029) (621,1030) (621,1030) (621,1031)  
(621,1032) (620,1033) (619,1033) (617,1034) (616,1034) (615,1034) (614,1034)  
(612,1033) (611,1033) (611,1032) (610,1031) (610,1030) (610,1030) (609,1029)  
(609,1028);

POLYGON (645,1030) (645,1031) (645,1032) (644,1032) (644,1033) (643,1033)  
(642,1034) (640,1034) (638,1034) (637,1033) (636,1033) (636,1033) (636,1032)  
(636,1032) (635,1031) (635,1031) (636,1030) (636,1030) (636,1029) (637,1029)  
(638,1029) (640,1028) (641,1028) (642,1028) (643,1028) (643,1028) (644,1027)  
(645,1027) (646,1027) (647,1026) (648,1025) (649,1024) (649,1023) (650,1022)  
(650,1021) (650,1019) (650,1018) (649,1017) (648,1016) (646,1014) (642,1014)  
(639,1014) (636,1014) (633,1016) (632,1016) (632,1017) (631,1017) (631,1018)  
(631,1019) (631,1020) (631,1021) (631,1022) (635,1022) (635,1021) (635,1020)  
(636,1019) (636,1019) (637,1018) (638,1017) (640,1017) (642,1017) (644,1018)  
(645,1018) (645,1019) (645,1019) (646,1020) (646,1020) (646,1021) (645,1021)  
(645,1022) (645,1022) (644,1022) (643,1023) (642,1023) (641,1024) (640,1024)  
(639,1024) (638,1024) (637,1025) (636,1025) (634,1026) (633,1026) (633,1027)  
(632,1028) (631,1030) (631,1031) (631,1032) (632,1033) (633,1035) (633,1036)  
(634,1036) (637,1037) (639,1038) (641,1038) (644,1038) (646,1037) (647,1036)  
(648,1034) (649,1033) (649,1031) (649,1030) (645,1030);

ESCAPE -802 ";

LINE (645,1030) (645,1031) (645,1032) (644,1032) (644,1033) (643,1033)  
(642,1034) (640,1034) (638,1034) (637,1033) (636,1033) (636,1033) (636,1032)  
(636,1032) (635,1031) (635,1031) (636,1030) (636,1030) (636,1029) (637,1029)  
(638,1029) (640,1028) (641,1028) (642,1028) (643,1028) (643,1028) (644,1027)

(645,1027) (646,1027) (647,1026) (648,1025) (649,1024) (649,1023) (650,1022)  
(650,1021) (650,1019) (650,1018) (649,1017) (648,1016) (646,1014) (642,1014)  
(639,1014) (636,1014) (633,1016) (632,1016) (632,1017) (631,1017) (631,1018)  
(631,1019) (631,1020) (631,1021) (631,1022) (635,1022) (635,1021) (635,1020)  
(636,1019) (636,1019) (637,1018) (638,1017) (640,1017) (642,1017) (644,1018)  
(645,1018) (645,1019) (645,1019) (646,1020) (646,1020) (646,1021) (645,1021)  
(645,1022) (645,1022) (644,1022) (643,1023) (642,1023) (641,1024) (640,1024)  
(639,1024) (638,1024) (637,1025) (636,1025) (634,1026) (633,1026) (633,1027)  
(632,1028) (631,1030) (631,1031) (631,1032) (632,1033) (633,1035) (633,1036)  
(634,1036) (637,1037) (639,1038) (641,1038) (644,1038) (646,1037) (647,1036)  
(648,1034) (649,1033) (649,1031) (649,1030) (645,1030);

ENDPIC;

ENDMF;

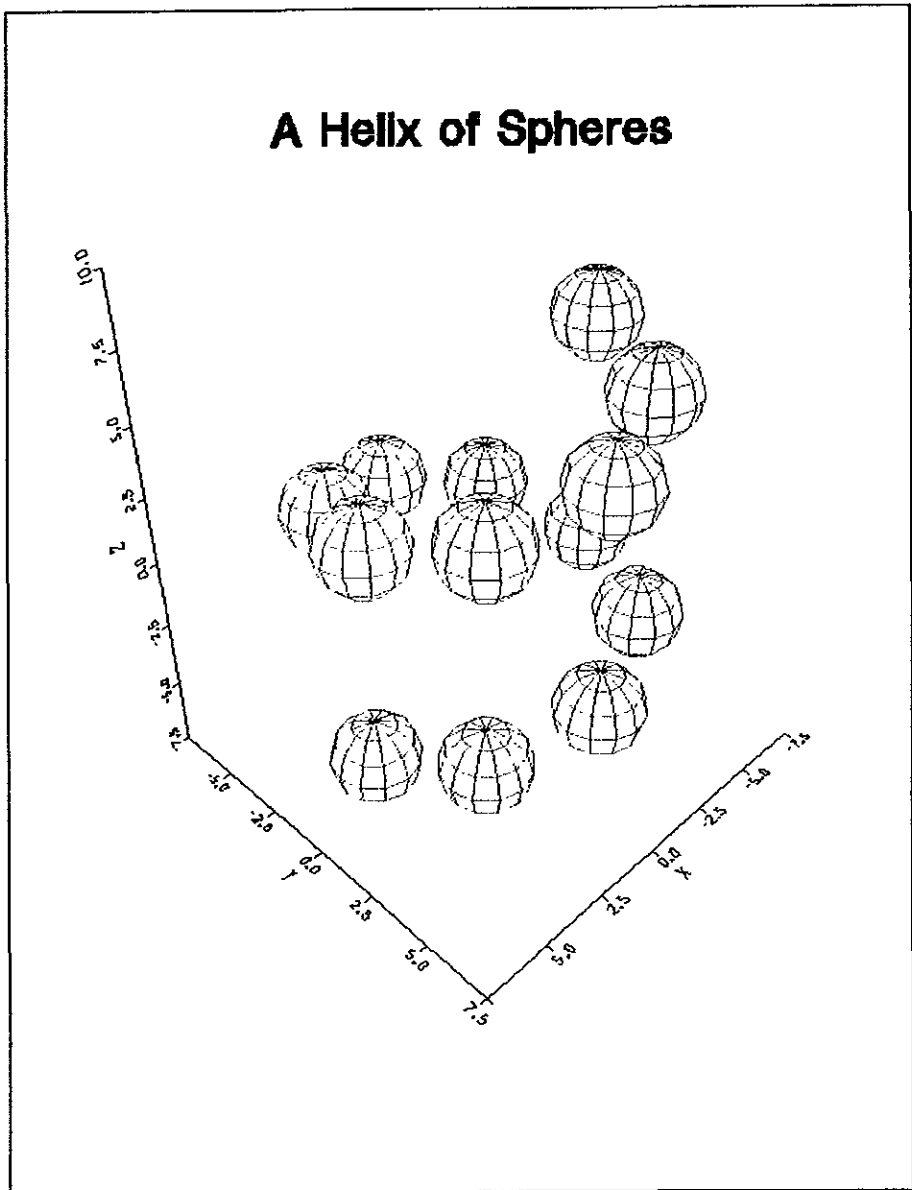


Figura 5.3. Vista del archivo HELIXSP CGM

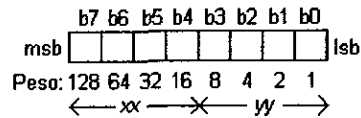
### 5.3.3 Metaarchivo en modo caracter.

La codificación en modo caracter está orientada principalmente hacia aquellas situaciones en donde es importante minimizar el tamaño del metaarchivo o transmitir este último a través de servicios de comunicación de tipo alfanumérico, es decir, orientados a caracteres. Esta codificación hace uso de una representación muy compacta de los datos, lo cual la hace idónea para el almacenamiento o transmisión entre diferentes sistemas de cómputo.

Esta modalidad de codificación no es muy utilizada por los programas de aplicación –a pesar de las ventajas de su reducido tamaño, en parte debido a la complejidad de su implementación. En estos casos la forma binaria es preferida debido a que reduce la sobrecarga de los sistemas de cómputo durante el proceso de decodificación de la imagen. En los siguientes párrafos se describirá brevemente su forma básica de codificación. Para una descripción más detallada de esta modalidad del CGM, se puede consultar la referencia técnica que aparece en la bibliografía al final de la tesis.

#### Códigos de operación.

Cada elemento gráfico es almacenado mediante uno o dos bytes de datos, divididos cada uno en dos campos de la forma  $xx/yy$  como se ilustra en la siguiente figura:



Donde se observa que  $xx$  es el número representado por los cuatro bits más significativos, mientras que el valor  $yy$  ocupa los primeros cuatro bits de datos. Cada componente de la pareja  $xx/yy$  puede almacenar un número comprendido entre 0 y 15. Basándose en las posibles combinaciones de estos valores, el estándar CGM en modo caracter define una relación de equivalencias (denominada *código de operación*) para cada elemento gráfico presente en un metaarchivo. La siguiente tabla muestra dichos códigos únicamente para el caso de aquellos elementos que serán utilizados por el sistema de conversión:

Nombre del elemento	Código(s) de operación
BEGIN METAFILE	03/0 02/0
END METAFILE	03/0 02/1
BEGIN PICTURE	03/0 02/2
BEGIN PICTURE BODY	03/0 02/3
END PICTURE	03/0 02/4
VDC TYPE	03/1 02/3

**Códigos de operación (continuación).**

---

Nombre del elemento	Código(s) de operación	
REAL PRECISION	03/1	02/4
BEGIN METAFILE DEFAULTS REPLACEMENT	03/1	02/11
END METAFILE DEFAULTS REPLACEMENT	03/1	02/12
SCALING MODE	03/2	02/0
VDC EXTENT	03/2	02/5
BACKGROUND COLOR	03/2	02/6
VDC REAL PRECISION	03/3	02/1
POLYLINE	02/0	
DISJOINT POLYLINE/	02/1	
POLYGON	02/6	
LINE WIDTH	03/5	02/2
LINE COLOUR	03/5	02/3
INTERIOR STYLE	03/6	02/1
FILL COLOUR	03/6	02/2
COLOUR TABLE	03/6	03/0



---

## **Parte III**

# **Desarrollo del sistema para la creación de archivos gráficos en formato CGM a partir de información sísmica**

---

## Capítulo 6

### Fase de definición

El desarrollo del software de conversión a formato CGM se describirá en estos dos últimos capítulos de la tesis, utilizándose para ello un enfoque de Ingeniería de Programación. Haciendo uso de una metodología de desarrollo que conduzca a la obtención de un programa que funcione eficientemente, se tratará de que éste sea fácil de comprender y posteriormente de darle mantenimiento correctivo por parte del Departamento de Programación y Sistemas.

Una visión general del desarrollo del sistema puede ser ilustrada mediante un diagrama que muestre las tres diferentes fases en que se divide el proceso de desarrollo de cualquier programa, independientemente del área de aplicación, tamaño del proyecto o complejidad.

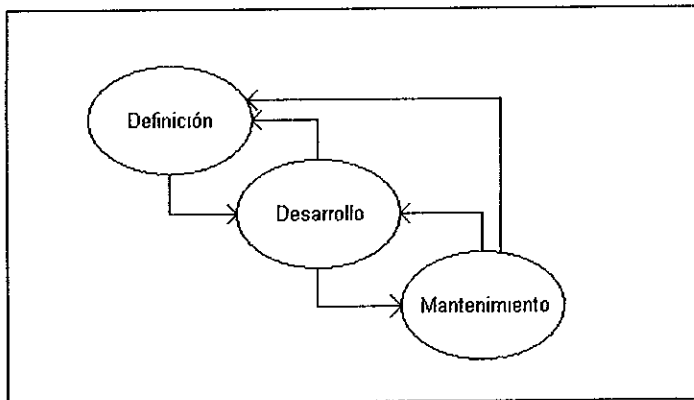


Figura 6.1. Visión generalizada del proceso de desarrollo del software.

Las tres fases, *definición*, *desarrollo* y *mantenimiento*, se encuentran en todos los proyectos de desarrollo de software. La fase de definición se enfoca sobre el *qué* del sistema, es decir, en esta etapa se establece la base del proyecto, identificando el tipo de información que ha de ser procesada, la función y el rendimiento deseado, el establecimiento de interfaces, las ligaduras de diseño, etc. Aunque los métodos a seguir durante esta fase variarán dependiendo del método de ingeniería de software aplicado de alguna forma se producirán tres pasos específicos:

**Análisis del sistema.** Define la función de cada elemento de un sistema informático, asignando finalmente el papel que jugará el software.

**Planificación del proyecto.** Una vez asignado el ámbito del software, se asignan los recursos, se estiman los costos y se definen las tareas y la planificación del trabajo.

**Análisis de requerimientos.** Representa el ámbito definido para el software de la dirección, pero antes de comenzar a trabajar, es necesario disponer de una información más detallada del dominio de la información y de la función del software.

Estos tres pasos se describirán en las siguientes secciones del presente capítulo.

La fase de desarrollo (descrita en detalle en el capítulo 7), se enfoca sobre el *cómo* del proyecto. Aquí el que desarrolla el software descubre cómo han de diseñarse las estructuras de datos y arquitectura del mismo, cómo han de implementarse los detalles procedimentales, cómo ha de trasladarse el diseño a un lenguaje de programación y cómo ha de realizarse la prueba. Los métodos aplicados durante esta fase variarán dependiendo de la metodología aplicada de ingeniería del software. Sin embargo, de alguna forma se producirán tres pasos concretos:

**Diseño.** Traslada los requerimientos del software a un conjunto de representaciones (algunas gráficas, otras tabulares o basadas en lenguajes) que describen la estructura de datos, arquitectura y procedimiento algorítmico.

**Codificación.** Las representaciones del diseño deben trasladarse a un lenguaje artificial que da como resultado unas instrucciones ejecutables por la computadora. El paso de la codificación ejecuta esta traslación.

**Prueba.** Una vez que el software se ha implementado en una forma ejecutable por la máquina, debe ser probado para descubrir los defectos que puedan existir en la función, lógica e implementación.

Por último la fase de mantenimiento se enfoca sobre el cambio asociado con una corrección de errores, adaptaciones requeridas por la evolución del entorno del software o modificaciones debidas a cambios en los requerimientos del cliente para reforzar o aumentar el sistema. La fase de mantenimiento reaplica los pasos de las fases de definición y desarrollo, pero en el contexto del software existente. Durante la fase de mantenimiento se encuentran tres tipos de cambios:

**Corrección.** Incluso con las mejores actividades para garantizar la calidad, es probable que el cliente descubra defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.

**Adaptación.** Con el paso del tiempo es probable que cambie el entorno original (por ejemplo, CPU, sistema operativo, periféricos) para el cual se desarrolló el software. El mantenimiento adaptativo se traduce en modificación del software para acomodarlo a los cambios de su entorno externo.

**Aumento.** Conforme se utiliza el software, el cliente reconocerá funciones adicionales que podría ser beneficioso añadirlas. El mantenimiento perfectivo aumenta el software más allá de sus requerimientos funcionales originales.

**Metodología de desarrollo para el sistema de conversión de formatos.**

Durante las pasadas cuatro décadas en que el desarrollo de software para computadoras evolucionó hasta adquirir un enfoque de ingeniería, se han propuesto varias metodologías (llamadas también paradigmas por algunos autores) para su desarrollo. El empleo de un método en particular representa un enfoque disciplinado en el proceso de desarrollo del software, el cual provee de guías para simplificar posteriormente la tarea de mantenimiento, cuando se requieran hacer modificaciones de tipo correctivo, adaptativo o perfectivo. Para el desarrollo del sistema de conversión de formatos se empleará el paradigma del ciclo de vida clásico para la ingeniería del software, algunas veces llamado el “modelo en cascada”.

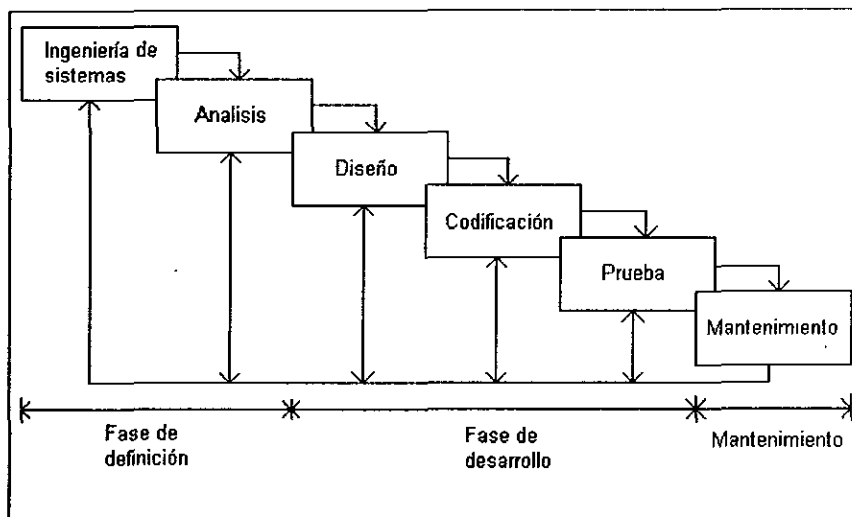


Figura 6 2 El ciclo de vida clásico.

Bajo este esquema se observa que las fases genéricas descritas anteriormente se definen explícitamente, ya que este paradigma exige un enfoque sistemático y secuencial en el desarrollo del software. Este comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento

La figura 6.3 ilustra los pasos genéricos que se deben realizar y las distintas representaciones del software que se derivan durante la fase de definición. Ya que el software es siempre parte de un sistema mayor basado en computadoras, la ingeniería de sistemas debe desarrollarse antes (o en conjunción con) la planificación. Durante esta etapa se desarrolla una descripción bien delimitada del alcance del esfuerzo, se definen los recursos necesarios para su desarrollo y se establecen las estimaciones de tiempo y costo. El propósito de la etapa de planificación es proveer de una indicación preliminar de la viabilidad del proyecto en relación al costo y a las restricciones de tiempo que ya se hayan establecido. Un documento denominado *Plan del Proyecto de Software* es realizado durante esta etapa por el analista del sistema.

El próximo paso en la fase de definición es el análisis y la definición de los requerimientos del software. En este paso el elemento del sistema asignado al software se define en detalle. Los requerimientos se analizan y se definen de una de dos maneras: el análisis formal del dominio de la información puede ser usado para establecer una representación del flujo de la información y de su estructura. Estas representaciones se expanden luego para convertirse en una especificación del software. En caso de que los requerimientos no sean conocidos, se construye un prototipo del software y es evaluado por el cliente en un intento de afianzar los requerimientos.

La tarea de análisis y definición de los requerimientos es un esfuerzo conjunto guiado por el diseñador del software (el analista de sistemas en este caso) y el cliente. La *Especificación de Requerimientos del Software* es un documento producido como resultado de esta etapa.

Finalmente, la fase de definición culmina con una revisión técnica del anterior documento (o en lugar de la Especificación, un prototipo del programa) por el diseñador y el cliente. Una vez que se han definido los requerimientos, el Plan del Software es reevaluado para su corrección. La información no cubierta durante el análisis de requerimientos puede modificar las estimaciones hechas durante la planificación. En las siguientes secciones se describe la realización de la fase de desarrollo para el caso del programa de conversión de formatos.

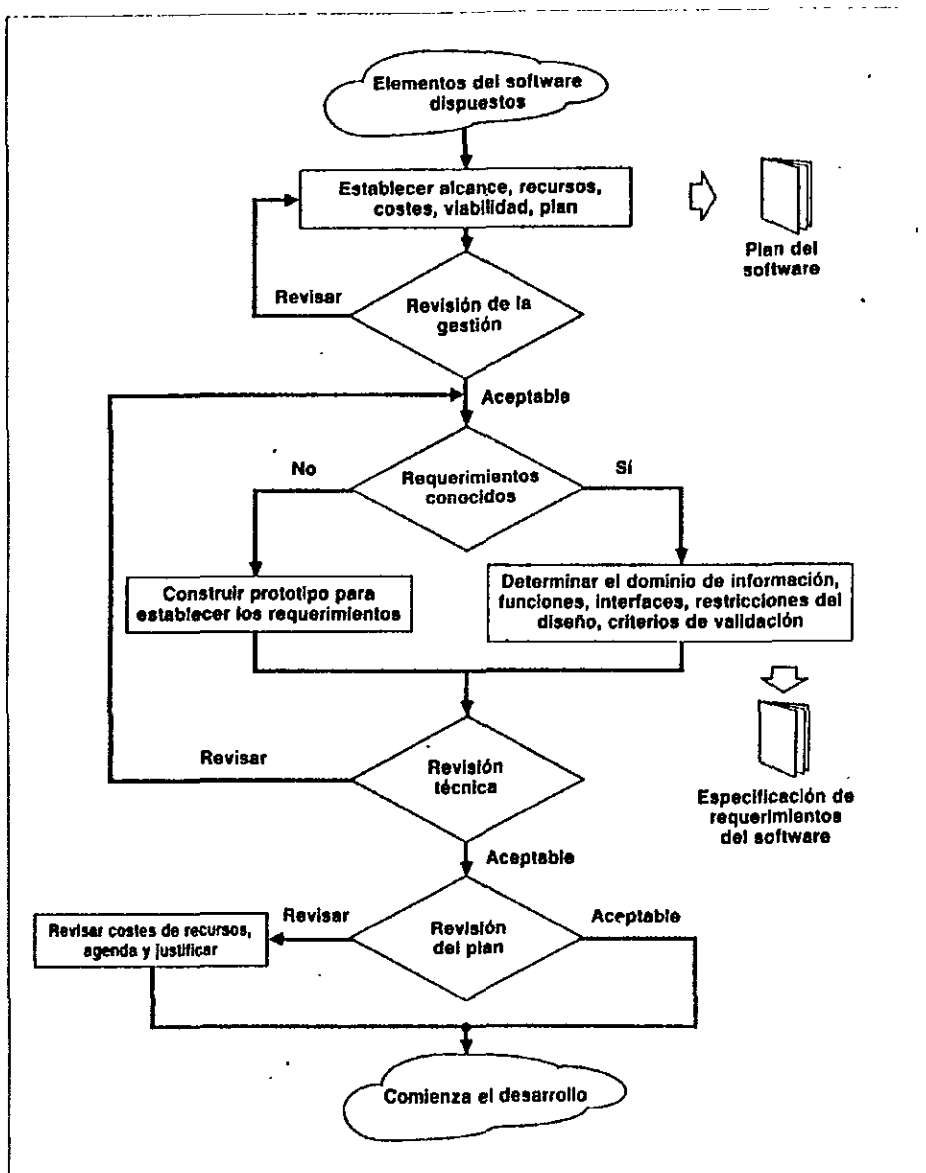


Figura 6.3. La fase de definición

## 6.1 Análisis del sistema.

Todo desarrollo de software está orientado hacia la resolución de un problema dentro de un contexto más amplio que involucra otros elementos que necesariamente interactúan con el mismo. El análisis del sistema, como primera actividad dentro de todo el proceso de desarrollo se centra justamente en el estudio de dichos elementos (no solo en el software a desarrollar) y que juntos dan forma a lo que se denomina *sistema basado en computadora*.

De esta manera, un sistema basado en computadora puede ser definido como *un conjunto u ordenación de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el procesamiento de la información*

Dichos elementos se muestran en la figura 6.4 e incluyen los siguientes:

**Software.** Los programas de computadora, estructuras de datos y documentación asociada que sirven para realizar el método lógico, procedimiento o control requerido.

**Hardware.** Los dispositivos electrónicos (CPU, memoria, etc.) que proporcionan la capacidad de computación y los dispositivos electromecánicos (sensores, motores, bombas, medios de transporte de cinta magnética, etc.) que proporcionan las funciones del mundo exterior.

**Gente.** Los individuos que son usuarios y operadores del software y del hardware.

**Bases de datos.** Una colección grande y organizada de información a la que se accede mediante el software y que es una parte integral del funcionamiento del sistema.

**Documentación.** Los manuales, impresos y otra información descriptiva que explica el uso y/o la operación del sistema.

**Procedimientos.** Los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

Una característica compleja de los sistemas basados en computadoras es que los elementos que componen un sistema pueden también representar un *macro elemento* de un sistema todavía mayor. Un macro elemento es un sistema basado en computadora que forma parte de un sistema basado en una computadora mayor. Por lo tanto, el papel del Ingeniero de Sistemas en esta primera etapa en el desarrollo del software, es el de definir los elementos de un sistema basado en computadora específico dentro del contexto de toda la jerarquía de sistemas (macro elementos).

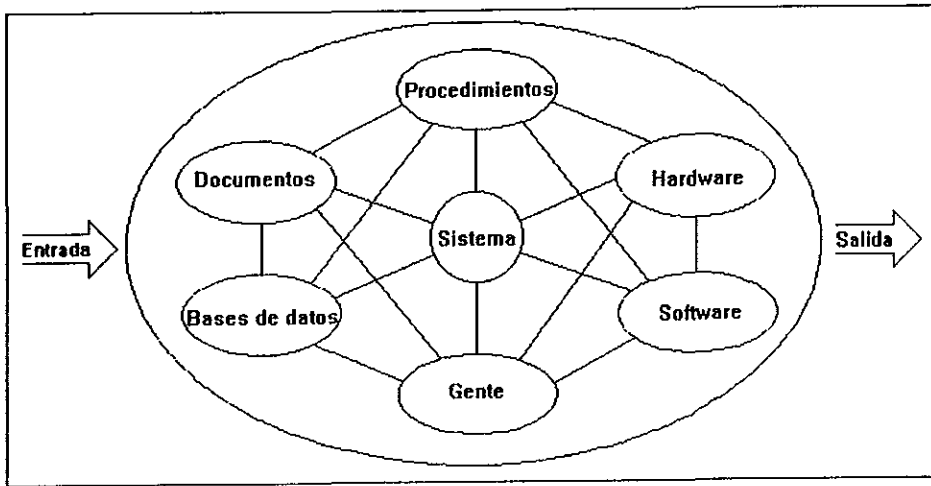


Figura 6.4. Elementos que intervienen en un sistema basado en computadora.

La génesis de un nuevo sistema basado en computadora comienza con una explicación preliminar de los objetivos que se desean para el mismo. Esta descripción es dada por parte del cliente al analista de sistemas, el cual a su vez debe *delimitar* el sistema identificando el alcance de funcionamiento y rendimiento deseados. Para el caso particular del programa a desarrollar en la presente tesis, de aquí en adelante se hará referencia al proyecto como *Sistema para la generación de gráficas en formato CGM a partir de CAPS (CAPS2CGM)*.

Como ingenieros de sistemas se nos presenta el siguiente conjunto de objetivos que debe cubrir el programa:

El CAPS2CGM es un programa de computadora que deberá ser desarrollado para procesar un flujo de información gráfica generado con el auxilio de un programa de procesamiento y graficado de información sísmica denominado CAPS (Computer Aided Plotting System: Sistema de Graficación Asistido por Computadora) en un computador central IBM 9121, con el propósito de efectuar su conversión a un formato gráfico vectorizado, estándar dentro de la industria e independiente de cualquier plataforma de hardware denominado CGM (Computer Graphics Metafile: Metaarchivo para Gráficos por Computadora). Las gráficas así obtenidas se deberán manejar como archivos independientes en disco o cinta magnética.

En esta primera etapa del proyecto se deben plantear y responder una serie de preguntas referentes al funcionamiento y rendimiento globales de todo el sistema, así como del flujo y contenido de la información que se va a procesar.



Entre las muchas preguntas que se podrían cuestionar y responder para el sistema a desarrollar están las siguientes:

1. En términos generales ¿Cuál es la forma en que opera la computadora principal IBM 9121?

Esta computadora es una máquina orientada al procesamiento por lotes (batch processing) en donde todos los programas, datos, subrutinas, procedimientos y órdenes al sistema operativo son agrupados en una única entidad denominada *trabajo* (job). Este consiste en un conjunto de *tarjetas* (cards) procesadas en forma secuencial y que describen cada operación que será ejecutada en la computadora. Hasta la década de los setenta, dichas tarjetas eran físicamente pequeños recuadros de cartoncillo con una estructura predefinida, en donde el usuario perforaba a mano o con ayuda de máquinas especiales (card punchers) la secuencia de instrucciones que componían el programa que deseaba ejecutar. Posteriormente un conjunto de estas tarjetas, las cuales daban forma a un *trabajo* eran ingresadas a la computadora por medio de dispositivos especiales denominados lectoras de tarjetas perforadas (card readers).

Aunque esta forma de ingresar datos a la computadora ha desaparecido por completo, el término *tarjeta* ha permanecido vigente hasta nuestros días, de manera que los *trabajos*, en lugar de ser físicamente grupos de tarjetas perforadas, hoy en día se representan mediante entidades lógicas en disco denominadas *miembros* (members) y que son una especie de archivos dentro del sistema operativo en el que opera esta computadora, denominado MVS (Multiple Virtual Storage).

Finalmente, por medio de un grupo de terminales conectadas al dispositivo controlador IBM 3174 11L (ver figura 4.1, capítulo 4 página 31), se ejecuta un programa de interfaz con el usuario denominado ISPF/PDF (Interactive System Productivity Facility/Program Development Facility) el cual es un ambiente de ventanas que representa una extensión del sistema operativo MVS conocida como TSO (Time Sharing Option: Opción de Tiempo Compartido). Esto permite que varios usuarios puedan interactuar con la computadora al mismo tiempo, permitiéndoles crear, editar, compilar o enviar para su ejecución *trabajos* a la computadora a partir de *miembros* almacenados en disco. Cabe aclarar que mediante este esquema la comunicación nunca se establece directamente con la computadora principal sino con el dispositivo controlador de terminales.

Esta computadora junto con el sistema paralelo IBM SP2 9076 constituyen las dos plataformas de cómputo principales que se utilizan en la Gerencia de Prospección Geofísica para el análisis y procesamiento de información sísmica.

2. ¿Qué significa CAPS y a grandes rasgos cómo trabaja esta aplicación?

CAPS es el acrónimo de Computer Aided Plotting System (Sistema de Graficación Asistido por Computadora). Básicamente es un programa auxiliar para el analista geofísico, el cual le permite crear gráficas de secciones sísmicas junto con algunos datos adicionales definidos directamente por el usuario. Este programa fue desarrollado por una compañía norteamericana de exploración geofísica: la Western Geophysical Company of America, para ser ejecutado en un ambiente de tipo batch (lotes), tal y como es el caso de la computadora principal IBM 9121.

Esta aplicación puede ser visualizada como un sistema de dos capas. En el nivel más alto (Plot Creation), tenemos un subsistema orientado a comandos (command driven system) que permite al usuario diseñar sus propias gráficas, independientemente de su disposición y tipo, por medio de la creación de unos archivos especiales llamados *esqueletos gráficos* (plot skeletons) que contienen comandos de graficación CAPS. Dichos comandos sirven principalmente para delimitar la región de trabajo donde se hará la gráfica, darle la orientación deseada e insertar texto descriptivo en diferentes tamaños y orientaciones.

El segundo nivel (Plot adaptation), está constituido por un conjunto de elementos de imagen prediseñados llamados *entidades gráficas* (plot entities) tales como marcadores, logotipos, símbolos especiales, escalas, etc., y que son invocados a partir de los esqueletos gráficos y añadidos a la gráfica final de salida.

Cada una de las dos anteriores capas se divide a su vez en varios pasos que corresponden directamente a los módulos de CAPS que se ejecutan durante un proceso. La tabla 6.1 muestra su secuencia de ejecución en la generación de una gráfica de sección sísmica.

Paso	Etapas	Módulo ejecutable
Creación (Plot Creation)	Composición (Plot Compositor)	Editor –Paso 1 (Plot Editor), S925
	Codificación (Plot Encoding)	Generador –Paso 2 (Plot Generator), S926
Adaptación (Plot Adaptation)	Formateo (Plot Formatting)	Conformador –Paso 3 (Plot Formatter), S928
	Conversión y mezcla [Plot Conversión & Overlay (mixing)]	Conversor –Paso 4 (Plot Compositor), S929

Tabla 6.1. Pasos en la generación de una gráfica de sección sísmica por medio de CAPS.

Como se puede observar, en total se requieren cuatro pasos para la obtención de una gráfica de sección sísmica. En el último de ellos (Plot Compositor) los datos sísmicos de entrada, almacenados en disco o cinta magnética, se añaden a la estructura de la gráfica que previamente ha sido definida en los archivos de esqueletos gráficos.

3. ¿Cuál es el proceso que se sigue en la *mainframe* para obtener una gráfica de sección sísmica por medio de CAPS?

El proceso completo para la obtención de una gráfica de sección sísmica comienza en la *mainframe* IBM 9121 con la ejecución de un proceso en CAPS (ver figura 6.5). Este programa es ejecutado según los pasos descritos en la pregunta anterior. Como se puede observar en la figura, la información gráfica que se genera al término del cuarto paso se transfiere como un flujo de datos a un equipo de cómputo denominado Rasterizador de Alta Velocidad HSR-11B (High Speed Rasterizer) de la firma AMF Logic Sciences Inc. Este subsistema opera como una interfaz en hardware que prepara los datos para ser graficados en un equipo de impresión electrostático. Dicho equipo es en este caso un graficador Xerox Versatec 6410.

Todo este proceso es transparente para el usuario, de modo que una vez ejecutado un programa en CAPS, el resultado del proceso es la obtención de una gráfica de sección sísmica.

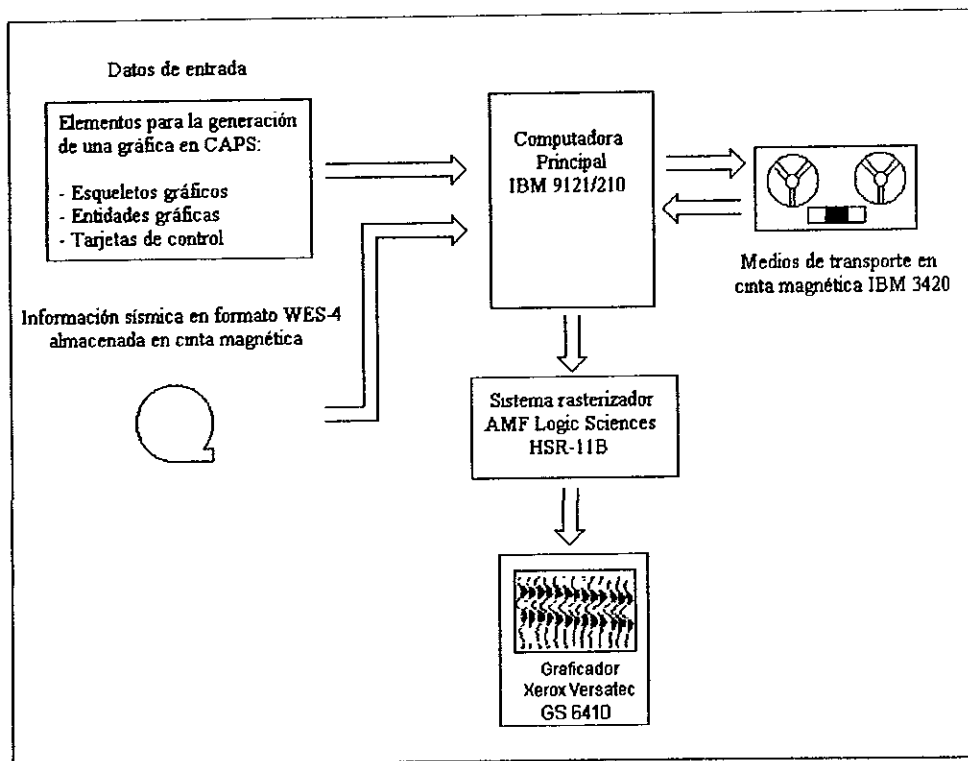


Figura 6.5. Flujo normal del procesamiento en CAPS.

#### 4. ¿Qué estructura tienen los datos que conforman las gráficas generadas por CAPS?

Estos datos tienen una estructura de metaarchivo en forma vectorial y dispuestos en un formato especial para ser procesados por el equipo rasterizador HSR-11B. La disposición de esta información se hace conforme a los diferentes comandos que el anterior sistema es capaz de procesar, los cuales fueron descritos en detalle en el capítulo 4.

#### 5. ¿Cuál es la misión del equipo rasterizador HSR-11B?

Convertir a un mapa de bits la información gráfica de la sección sísmica que le entrega el computador central al término de la ejecución de un proceso en CAPS. El propósito de hacer esta conversión se debe a las características propias del equipo de graficación que se emplea —en caso un graficador Xerox Versatec 6410, el cual solo puede imprimir imágenes tipo *raster*, es decir, este equipo no hace uso de plumillas para la copia impresa de las secciones sísmicas.

#### 6. Al término de un proceso en CAPS ¿los datos que componen la gráfica pueden retenerse en algún sitio, por ejemplo, un archivo en disco o cinta magnética?

Sí, esto se logra mediante una pequeña modificación en las tarjetas de control que forman el *trabajo* correspondiente a un proceso en CAPS (ver nuevamente la figura 6.5) En estas tarjetas se redireccionan los datos que conforman la gráfica al final del cuarto paso, de manera que en vez de dirigirse al equipo rasterizador, se guarden en un archivo independiente en disco o cinta.

#### 7. ¿El desarrollo del programa de conversión debe hacerse en el mismo entorno en donde se generan las gráficas o existen plataformas de desarrollo alternativas?

No necesariamente, una opción alternativa pueden ser los equipos IBM de tecnología RISC 6000 (por ejemplo uno de los nodos del sistema SP2 9076). Hay que recordar que en estos equipos se ejecutan también otros programas para el procesamiento de información geofísica, y que las gráficas en formato CGM que se desea generar son independientes de cualquier plataforma de hardware. Dadas estas circunstancias, sólo haría falta diseñar una interfaz entre ambas computadoras para transferir la información gráfica de las secciones sísmicas.

### 6.1.1 Asignaciones.

Partiendo de las respuestas anteriores, el siguiente paso consiste en desarrollar un número de asignaciones alternativas, esto es, se discutirán los métodos, caminos o procedimientos mediante los cuales puede resolverse el problema. Hay que notar que en este caso la función del sistema es simplemente la implementación de un procedimiento secuencial para el manejo de datos, por lo tanto, el rendimiento no está definido explícitamente. Con la información así obtenida, y haciendo un estudio de las dos posibles plataformas de desarrollo disponibles, se proponen las siguientes alternativas de solución:

### Alternativa 1.

Se desarrolla un software de conversión en el mismo entorno en donde se generan las gráficas, es decir, en el computador central IBM 9121. La información gráfica de un proceso en CAPS se almacena como un *miembro* en disco, estos datos son posteriormente leídos por el programa encargado de la conversión y la gráfica resultante en formato CGM queda almacenada también como un *miembro* en disco.

### Alternativa 2.

El software de conversión se desarrolla en uno de los nodos del sistema paralelo IBM SP2 9076. El programa efectúa un proceso sobre un archivo almacenado en disco que contiene los datos de la gráfica de sección sísmica, dando por resultado una imagen en formato CGM que será almacenada también como un archivo en disco en el mismo sistema. Esta opción requiere el diseño de una interfase para la transferencia de la información gráfica generada por CAPS desde la computadora principal.

Evaluando las asignaciones alternativas para el CAPS2CGM, es claro que la misma función puede ser asignada a dos elementos diferentes de hardware del sistema. Para elegir la asignación más efectiva, se debe aplicar un conjunto de criterios de evaluación a cada una de ellas. Existen varios criterios que controlan la selección de una configuración del sistema basándose en una asignación específica. En este caso, un análisis técnico y de interfaz con el entorno nos señalan las siguientes razones que justifican el desarrollo del sistema según lo especificado en la alternativa 2.

- Para el momento en que se hace este desarrollo, el Centro de Proceso de la Gerencia de Prospección Geofísica atraviesa por una etapa de reestructuración en sus equipos de cómputo, el cual consiste en un cambio de plataforma del equipo IBM 9121 al sistema paralelo IBM SP2 9076. El tiempo de vida útil de la *mainframe* es de aproximadamente 1 año, con lo cual el programa de conversión ya no estaría disponible en el momento en que se diera de baja dicho equipo si su desarrollo se hace según lo expuesto en la alternativa 1.
- CAPS es una aplicación que se encuentra disponible para varias plataformas de sistema operativo. Aparte de la versión para MVS en *mainframes* de IBM, como es el caso de la computadora 9121, existen versiones para CMS (Conversational Monitor System), VM (Virtual Machine) y UNIX. Los tres primeros son sistemas operativos de IBM para sus equipos de gran capacidad de cómputo. UNIX en cambio, es un sistema operativo disponible para una amplia variedad de equipos, no solo de IBM sino de otras marcas.

El sistema SP2 9076 opera sobre una versión propietaria de UNIX para computadoras de IBM denominado AIX. El desarrollo del sistema de conversión sobre esta plataforma de sistema operativo haría que el programa de conversión fuera transportable a nivel binario entre computadoras de la familia IBM RISC 6000, al cual pertenece el sistema SP2, dado que estos equipos son los que se utilizan en la mayoría de los Centros de Proceso de Pemex.

El entorno de OMEGA –programa para el procesamiento de información sísmica desarrollado también por la compañía Western Geophysical, que se ejecuta en el sistema paralelo IBM SP2 9076, incluye precisamente una versión de CAPS, de esta manera, el HSR2CGM se podrá seguir utilizando aun después de que se haya dado de baja la computadora principal, ya que incluso los equipos electrostáticos de graficación que se utilizan en el computador central también pueden ser usados por el sistema paralelo.

Una vez hecho lo anterior y recordando los elementos genéricos que intervienen en un sistema basado en computadora (figura 6.4), se observa que para la alternativa 2 dichos elementos quedan finalmente asignados de la siguiente manera:

**Hardware.** Representado en este caso por el sistema paralelo IBM SP2 9076, lugar donde se hará el desarrollo y se ejecutará finalmente el programa de conversión de formatos.

**Software.** El programa a desarrollar encargado de generar las gráficas

**Gente.** El grupo de analistas geofísicos que harán uso del sistema.

**Bases de datos.** Incluida dentro del mismo software de conversión como una tabla de elementos gráficos y la forma en que son codificados, representa en este caso toda la información que el programa debe acceder para generar un archivo en formato CGM.

**Documentación.** Las distintas representaciones del software (plan del proyecto, especificación de requerimientos, diseño, estructuras de datos, listado y especificación de pruebas) que se derivan según se pasa del concepto a la realización. Todos estos elementos son los que se describen en este y en el próximo capítulo.

**Procedimientos.** El manual de operación del programa, incluido como un documento que acompaña al software de conversión.

## 6.2 Esquema del plan del proyecto de software.

La planificación del proyecto de software se da como consecuencia de la asignación de la función y el rendimiento llevada a cabo como parte de la ingeniería de sistemas de computadoras. Durante la planificación se establece el alcance del esfuerzo para el desarrollo del software, y se estiman los recursos, costos y la planificación temporal.

### 6.2.1 Alcance.

#### a) Objetivo.

Desarrollo de un programa encargado de procesar el flujo de información gráfica generado en un computador central IBM 9121 mediante un programa para el procesamiento y graficado de información sísmica denominado CAPS (Computer Aided Plotting System: Sistema de Graficación Asistido por Computadora), con el propósito de efectuar su conversión a un formato gráfico vectorizado, estándar dentro de la industria e independiente de cualquier plataforma de hardware denominado CGM (Computer Graphics Metafile).

#### b) Funciones principales.

Creación de archivos gráficos en formato CGM en sus tres formas de codificación: binaria, en modo de texto y en modo carácter de las gráficas de secciones sísmicas generadas por CAPS en la computadora principal.

#### c) Otras características.

El programa de conversión incluirá opciones para la codificación de las trazas sísmicas en el archivo de salida según las modalidades de traza simple (*wiggle*), área variable y traza simple con área variable. Las dos últimas opciones tienen contemplado un relleno positivo y negativo de los registros sísmicos.

#### d) Un escenario de desarrollo.

El desarrollo se llevará a cabo en una computadora IBM SP2 9076, bajo el ambiente del sistema operativo AIX (una versión propietaria de UNIX para máquinas IBM de tecnología RISC) y en lenguaje de programación ANSI C.

## 6.2.2 Recursos.

La estimación de los recursos necesarios para acometer el esfuerzo de desarrollo del software se puede visualizar como una pirámide, en la base deben existir las herramientas hardware y software –que soporten el esfuerzo de desarrollo. En el nivel más alto, siempre se requiere el recurso primario: la gente.

Cada recurso generalmente se especifica con cuatro características: descripción del recurso (o habilidad requerida en el caso del recurso gente), informe de disponibilidad, fecha cronológica en la que se requiere el recurso y tiempo en que será aplicado el recurso. Las dos últimas características pueden ser vistas como una ventana temporal, es decir, el lapso de tiempo en que se tendrá acceso a las herramientas de hardware y software dentro de la organización donde se hará el desarrollo. La disponibilidad del recurso para una ventana temporal específica tiene que ser establecida lo más pronto posible.

### a) Recursos humanos.

Dado que el proyecto es relativamente pequeño (una persona-año o menos), un solo individuo se encargará de llevar a cabo todos los pasos del proceso de desarrollo del software, consultando con especialistas siempre que lo requiera.

### b) Recursos de hardware.

Se deben considerar en este caso tres categorías: el *sistema de desarrollo*, es decir, la computadora y los periféricos asociados que se utilizarán durante la fase de desarrollo del software; la *máquina objetivo* en la cual el software será ejecutado eventualmente y otros elementos de hardware del nuevo sistema.

Para el desarrollo del HSR2CGM la máquina objetivo y el sistema de desarrollo son idénticos, representados en este caso por el sistema paralelo IBM SP2 9076. El computador central IBM 9121 puede considerarse como un elemento del hardware externo en el desarrollo del nuevo sistema donde únicamente se generará el archivo de información sísmica que será procesado.

### c) Recursos de software.

Divididas en *herramientas orientadas al código* y *herramientas de metodología*, las primeras están representadas por el compilador del lenguaje de programación C denominado *xlc* en el ambiente del sistema operativo AIX y el editor de texto en pantalla *vi*, ambos en la computadora IBM SP2 9076.

Las herramientas de metodología dan soporte a la planificación del proyecto, al análisis de los requerimientos, al mantenimiento y a otras actividades. En este caso se utilizó la versión 98 de Microsoft Project en el ambiente de una computadora personal de escritorio tipo PC y bajo el sistema operativo Windows 95.



#### d) Ventanas de disponibilidad.

Se refiere al período de tiempo en el que estarán disponibles o se tendrá acceso a los recursos anteriores. Este lapso abarcará todo el tiempo que se tiene contemplado en el desarrollo del sistema y que es de aproximadamente cuatro meses (ver diagrama de Gantt de la página siguiente).

#### 6.2.3 Costos.

Dado que el desarrollo del sistema HSR2CGM responde a una propuesta hecha por parte de la Gerencia de Prospección Geofísica del IMP como parte de un programa de titulación denominado "Programa maestro para la formación de recursos humanos para la Industria Petrolera" dentro del mismo Instituto, un estudio de viabilidad de costos no es en este caso un factor importante, ya que no se trata de un desarrollo independiente que vaya a ser comercializado de alguna forma.

#### 6.2.4 Plan temporal.

La planificación temporal para proyectos de desarrollo de software puede ser vista desde dos perspectivas diferentes. En la primera, la fecha final de lanzamiento del sistema basado en computadora ya ha sido establecida. La organización encargada del desarrollo del software se ve forzada a distribuir el esfuerzo dentro del marco prescrito. El segundo enfoque de la planificación temporal del software asume que se han discutido unos límites cronológicos aproximados, por lo que la fecha final es fijada por la misma organización encargada del desarrollo del software. El esfuerzo se distribuye para hacer un mejor uso de los recursos y la fecha final se define después de un cuidadoso análisis del elemento del software.

Para el desarrollo del HSR2CGM se estableció el primer enfoque, dado que los límites de tiempo para la entrega del sistema fueron fijados de antemano por parte de las autoridades del Instituto en seis meses como máximo partir de la aprobación del proyecto por parte de un comité dictaminador.

Por lo tanto para la distribución de esfuerzos en el desarrollo del software se siguió un modelo empírico conocido como *regla 40-20-40* que enfatiza el inicio de las tareas de análisis y diseño y el final de las pruebas, distribuyéndose en cuatro meses a partir de la etapa de la Ingeniería de Sistemas y los restantes dos para la fase de mantenimiento del software. La distribución del esfuerzo realizado se ilustra en el siguiente diagrama de Gantt.

		3 Aug 97			10 Aug 97			17 Aug 97							
		F	S	M	T	W	T	F	S	M	T	W	T	F	S
1	Task Name Fase de definición														
		Fase de definición													
2	2 Análisis del sistema														
		Análisis del sistema													
3	3 Planificación del proyecto de software														
		Planificación del proyecto de software													
4	4 Análisis de requerimientos														
		Análisis de requerimientos													
5	5 Fase de desarrollo														
		Fase de desarrollo													
6	6 Diseño del software														
		Diseño del software													
7	7 Codificación														
		Codificación													
8	8 Prueba														
		Prueba													
9	9 Fase de mantenimiento														
		Fase de mantenimiento													

Task		External Tasks	
Progress		Project Summary	
Milestone			
Summary			

Proyecto: HSR2CGM  
Miguel Angel Avila Martinez  
Fecha: 17/Junio/1998

24 Aug '97	31 Aug '97	7 Sep '97	14 Sep '97	21 Sep '97	28 Sep '97	5 Oct '97	12
S M T W T F S S	M T W T F S S M	T W T F S S M T	S M T W T F S S M T	F S S M T W T F S S M T	W T F S S M T W T F S S	W T F S S M T W T F S S	

20 days

8 days

15 days

Fase

Diseño del software

Codificación



External Tasks  
Project Summary



Rolled Up Task  
Rolled Up Milestone  
Rolled Up Progress



Task  
Progress  
Milestone  
Summary



Split

Proyecto: HSR2CGM  
Miguel Angel Avila Martinez  
Fecha: 17/Julio/1998



### 6.3 Análisis de requerimientos.

La tarea de análisis de requerimientos es el primer paso técnico en el proceso de la ingeniería del software. En este punto se establece en forma general el ámbito del programa y este se da en una especificación concreta que se convierte en la base para la fase de desarrollo.

Un documento denominado *Especificación de Requerimientos del Software* se genera durante esta etapa para refinar la función y comportamiento asignados al software como parte de la ingeniería de sistemas. Para ello se establece una descripción completa de la información, una descripción funcional detallada, una indicación de los requerimientos de rendimiento y las ligaduras de diseño, unos criterios de validación apropiados y otros datos pertinentes a los requerimientos.

#### 6.3.1 Descripción de la información.

El siguiente diagrama de flujo de datos (DFD) describe el sistema de conversión de formatos considerándolo como un sistema basado en computadora. Este diagrama ilustra el flujo y las transformaciones que se aplican a la información gráfica de las secciones sísmicas desde que es generada en el computador central IBM 9121 hasta que se obtiene una imagen en formato CGM en el sistema paralelo IBM SP2 9076.

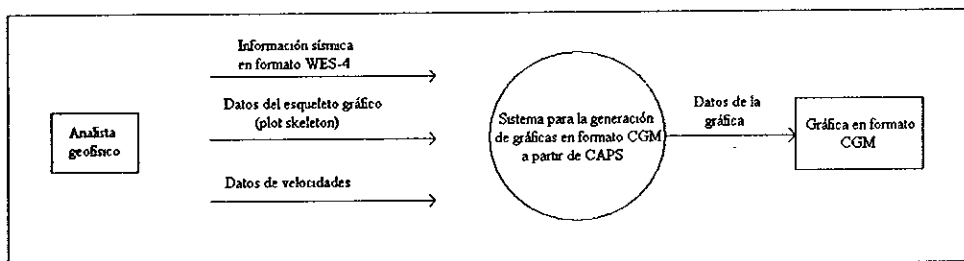


Figura 6.6. Diagrama de flujo de datos del sistema de conversión de formatos. Nivel 0: modelo de sistema fundamental.

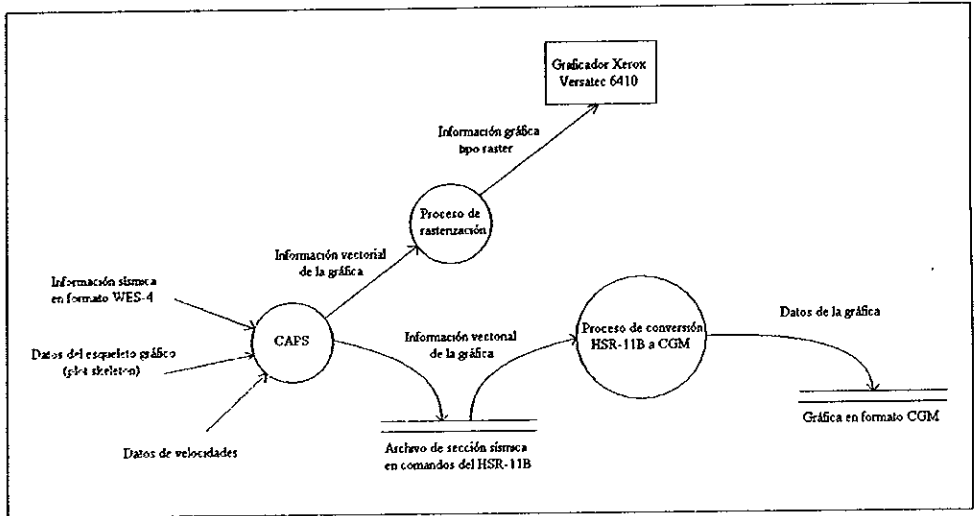


Figura 6.7. Diagrama de flujo de datos. Nivel 01

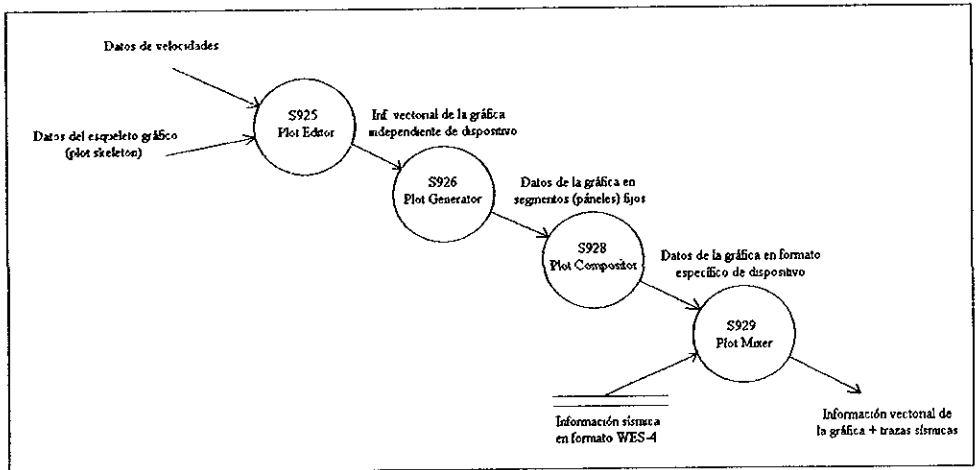


Figura 6.8. Diagrama de flujo de datos. Nivel 02. Refinamiento del CAPS en el computador central IBM 9121.

## Diccionario de datos.

Un análisis del dominio de la información puede ser incompleto si sólo se considera el flujo de datos. Cada flecha de un diagrama de flujo de datos representa uno o más elementos de información. Por lo tanto, se debe disponer de algún método para representar su contenido.

El diccionario de datos contiene las definiciones de todos los datos mencionados en el DFD, en una especificación del proceso y en el propio diccionario de datos. Los datos compuestos (datos que pueden además ser divididos) se definen en términos de sus componentes. Los datos elementales (datos que no pueden ser divididos) se definen en términos del significado de cada uno de los valores que pueden asumir. Por lo tanto, el diccionario de datos está compuesto de definiciones de flujo de datos, archivos (datos almacenados) y datos usados en los procesos (transformaciones).

Refiriéndonos al nivel 1 del DFD para el sistema de conversión de formatos, se observa que el elemento de datos *Información vectorial de la gráfica* se especifica como entrada al proceso principal que constituye el software que se va a desarrollar en la presente tesis. Pero ¿Qué es exactamente la *Información vectorial de la gráfica* en este contexto? El diccionario de datos nos da una definición precisa de este elemento de información para el DFD en cuestión:

Información vectorial de la gráfica	=	Comandos nativos del sistema rasterizador HSR-11B
Comandos	=	Command type + Word count + + {Datos del comando} <sup>n</sup>
Command type	=	[ No Operation   Seismic Traces   Timming Line   Serial Data   Seismic Flush   Restore Raster Buffer   Graphic Flush   Sectored Vectors   Scan Line   Advance   Initialize   Alphanumeric   Resampling]
No Operation	=	0
Seismic Traces	=	1
Timming Line	=	2
Serial Data	=	3
Seismic Flush	=	4
Restore Raster Buffer	=	5
Graphic Flush	=	6
Sectored Vectors	=	7
Scan Line Advance	=	8
Initialize	=	9
Alphanumeric	=	10
Resampling	=	11
Word count	=	Número entero positivo de 16 bits
Datos del comando	=	Número entero positivo de 16 bits.

Por otra parte, al término del proceso se obtienen los *datos de la gráfica* que componen el archivo en formato CGM de la sección sísmica. La representación dentro del diccionario de datos para este elemento es la siguiente:

Datos de la gráfica	=	{Elemento gráfico} <sup>n</sup>
Elemento gráfico	=	Encabezado + Datos del elemento
Encabezado	=	Clase + Número de elemento + Longitud
Clase	=	[Elementos delimitadores   Descripción de elementos del metaarchivo   Descripción de elementos gráficos   Elementos de control   Elementos primitivos de la gráfica   Atributos de los elementos   Elementos de escape   Elementos externos   Reservado]
Elementos delimitadores	=	0
Descripción de elementos del metaarchivo	=	1
Descripción de elementos gráficos	=	2
Elementos de control	=	3
Elementos primitivos de la gráfica	=	4
Atributos de los elementos	=	5
Elementos de escape	=	6
Elementos externos	=	7
Reservado	=	[8   9   10   ...   15]
Número de elemento	=	Número entero positivo.
Longitud	=	Número entero positivo.
Datos del elemento	=	{Byte} <sup>n</sup>



### 6.3.2 Criterios de validación.

Durante esta etapa se intenta responder básicamente a la pregunta de cómo reconocer una buena implementación del sistema. Para ello se definen una serie de pruebas que deben realizarse para validar la función que lleva a cabo el software desarrollado.

En el caso del HSR2CGM se requiere ante todo que las imágenes (archivos) de las secciones sísmicas codificadas en formato CGM y producidas en el sistema SP2 9076, sean una reproducción exacta en tamaño y apariencia de sus contrapartes generadas en el computador central IBM 9121. Esto es especialmente importante ya que las dimensiones y el nivel de resolución (detalle) de los registros sísmicos no debe modificarse ya que en tal caso implicaría la modificación de la información sísmica de entrada y por lo tanto la pérdida de los datos originales de campo.

Para verificar lo anterior, lo más sencillo es efectuar varias pruebas comparativas de los resultados que se obtienen en ambos sistemas. En primer lugar, se ejecuta un proceso en CAPS utilizando el computador principal y se imprime su resultado en uno de los graficadores electrostáticos –tal y como se ha hecho hasta ahora. Enseguida se ejecuta nuevamente el anterior proceso en CAPS en la *mainframe* pero en vez de imprimir su resultado, la información gráfica se guarda como un archivo independiente, por ejemplo en cinta magnética

Posteriormente este archivo (el cual contiene una representación de la gráfica de sección sísmica en comandos propios del HSR-11B), se copia en disco en uno de los nodos del sistema paralelo SP2 9076, donde a su vez sirve como entrada al programa de conversión de formatos que se encarga de generar un archivo equivalente en formato CGM

Finalmente, el archivo CGM generado se imprime en uno de los graficadores electrostáticos a través de utilerías propias del ambiente de los programas OMEGA o PROMAX. El resultado así obtenido puede compararse con la copia impresa generada directamente por el computador central, auxiliándose de un miembro del grupo de analistas geofísicos para determinar si la nueva sección sísmica es aceptable en su apariencia para los intérpretes.

### 6.3.3 Manual de usuario preliminar.

Esta última sección del análisis de requerimientos contiene información que complementa la especificación. El propósito del presente *Manual de Usuario Preliminar* es el de presentar al software como una caja negra. Esto es, se pone énfasis en la entrada del usuario y en la salida resultante. Esto puede servir posteriormente como una útil herramienta para descubrir problemas en la interface hombre-máquina

## HSR2CGM

### Manual de usuario preliminar

Programa para la conversión de archivos gráficos de secciones sísmicas en comandos nativos del sistema HSR-11B a formato CGM.

Este programa tiene como objetivo convertir a formato CGM (Computer Graphics Metafile: Metaarchivo para Gráficos por Computadora), las gráficas de secciones sísmicas generadas en la computadora principal IBM 9121 con el auxilio del programa CAPS (Computer Aided Plotting System).

El programa de conversión se encuentra disponible en el nodo 21 del sistema paralelo IBM SP2 9076 en la ruta `/usr/local/bin`. Para no tener que trasladarse a este subdirectorio, los usuarios pueden incluir la ruta del programa en su archivo de inicio de sesión `.profile` para Korn Shell o `.cshrc` para C-Shell, y así poder hacer uso del mismo desde cualquier lugar donde se encuentren dentro del sistema de archivos de AIX.

El sistema de conversión se utiliza como una línea de comando en la cual se debe teclear el nombre del programa, seguido de una serie de parámetros opcionales y/o necesarios de acuerdo a la siguiente sintaxis:

```
hsr2cgm [{-b|-t|-c}] fuente[.hsr] [destino] [presentación]
```

Donde

- b: conversión a formato cgm en modo binario (opción por defecto).
- c: conversión a formato cgm en modo carácter.
- t: conversión a formato cgm en modo de texto.

Si no se proporciona el nombre del archivo destino, este tendrá el mismo nombre que el archivo fuente pero con la extensión correspondiente al tipo de gráfica que se desee generar:

- .cgm para archivos binarios.
- .clt para archivos en modo de texto.
- .chr para archivos en modo carácter.

El parámetro opcional *presentación* permite forzar la forma de despliegue de la información sísmica en el archivo gráfico de salida, de acuerdo a una de las siguientes opciones.

Parámetro <i>presentación</i>	Significado
-vapfill	Área variable, relleno positivo
-vanfill	Área variable, relleno negativo
-wiggle	Únicamente el wiggle
-vawpfill	Área variable, wiggle, relleno positivo
-vawnfill	Área variable, wiggle, relleno negativo

Si se omite este parámetro las trazas sísmicas adoptaran la forma especificada en el archivo de entrada de comandos del HSR-11B.

Notas para la obtención de los archivos de comandos del sistema HSR-11B a partir de CAPS.

El programa CAPS es una herramienta utilizada por los analistas geofísicos en la computadora principal IBM 9121 para el procesamiento y graficado de información sísmica. El usuario de esta aplicación define por medio de *esqueletos gráficos* (plot skeletons), contenidos en uno o varios miembros, la estructura de la gráfica de salida. Los esqueletos gráficos son secuencias de instrucciones CAPS que describen imágenes creadas por el analista como sellos descriptivos, tablas de velocidades, de coordenadas, etc., y que junto con la información sísmica componen al final del proceso una gráfica de *sección sísmica*.

Este programa se ejecuta en cuatro pasos secuenciales:

- 1.-Plot Editor
- 2.-Plot Formatter
- 3.-Plot Compositor
- 4.-Plot Overlay

En el último paso (plot overlay), la información sísmica que se desea procesar, almacenada en disco o cinta, se combina con la definición de los esqueletos gráficos, dando por resultado una imagen de salida que normalmente se imprime en el graficador Xerox Versatec 6210.

El programa de conversión requiere que la información generada al final del cuarto paso, se procese como un archivo independiente en disco en el nodo 21 del sistema paralelo IBM SP2 9076 ya que el programa de conversión fue desarrollado y se ejecuta en esta plataforma, es decir, los datos que conforman la gráfica, antes de ser enviados al graficador Versatec deben ser guardados como un archivo en disco y ser copiados al anterior nodo. Una manera de lograr esto es almacenar en cinta magnética los datos de la gráfica al final del último paso de CAPS, y luego transferir esta información al disco duro del nodo 21 con ayuda de las utilerías de copiado presentes en el sistema operativo AIX tales como *tcopy*, *tcil* o *mt*.

Para transferir los datos que conforman la gráfica en CAPS hacia cinta magnética es necesario definir un nuevo archivo de salida *DATA2* en el módulo S929 (Plot Compositor), el cual se utilizará en forma alternada con el ya existente y que normalmente tiene asociado el valor *DUMMY*. Esto envía los datos de salida a cinta o directamente al graficador Versatec. Para lograr lo anterior se comenta en la línea de órdenes de JCL las líneas de *DATA2* que no se utilicen, como se muestra el siguiente ejemplo:

```

/* SALIDA AL GRAFICADOR VERSATEC
//DATA2 DD DUMMY
/* SALIDA A CARTUCHO 3480
/*DATA2 DD UNIT=(TAPE5,,DEFER),DCB=DEN=5,           C
/*          DSN='PR03.PRUEBA.CAPS',                 C
/*          VOL=SER=SALCAP,DISP=(NEW,KEEP),         C
/*          LABEL=(,BLP,EXPDT=98000)

```

La anterior disposición mandaría el resultado directamente al graficador. Para enviar el resultado a cinta, es necesario comentar el primer *DATA2* y quitar el asterisco de las cuatro últimas líneas. En este ejemplo la salida se enviaría a un cartucho tipo 3480.

Una vez hecho lo anterior, existen varios programas utilitarios dentro del sistema operativo AIX para el manejo y copiado de información en cinta magnética. Aquí el usuario debe hacer uso de un programa tal como *tcfl* o *mt* para copiar la información contenida en cinta a un archivo en disco en el nodo 21 y utilizando alguno de los transportes de cinta que están disponibles en el sistema SP2. El archivo que se genere podrá tener el mismo nombre que tenía el *miembro* de CAPS en el computador central, mas una extensión arbitraria que para propósitos de referencia dentro del programa de conversión será *.hsr*

Por último, se ejecuta el programa de acuerdo a la sintaxis descrita al inicio del presente manual. El archivo *CGM* que se genere quedará almacenado en un archivo en disco en el mismo nodo y en la ruta especificada por el usuario.

# Capítulo 7

## Fase de desarrollo

La fase de desarrollo traduce un conjunto de requerimientos en el elemento del sistema operacional que constituye lo que se denomina software de computadora. Una vez que se han establecido dichos requerimientos, la fase de desarrollo comprende tres pasos distintos: diseño, generación de código (manual o automáticamente) y prueba. Cada paso transforma la información de forma que finalmente se obtiene un software para computadora validado. El flujo de información durante esta fase se muestra en la figura 7.1.

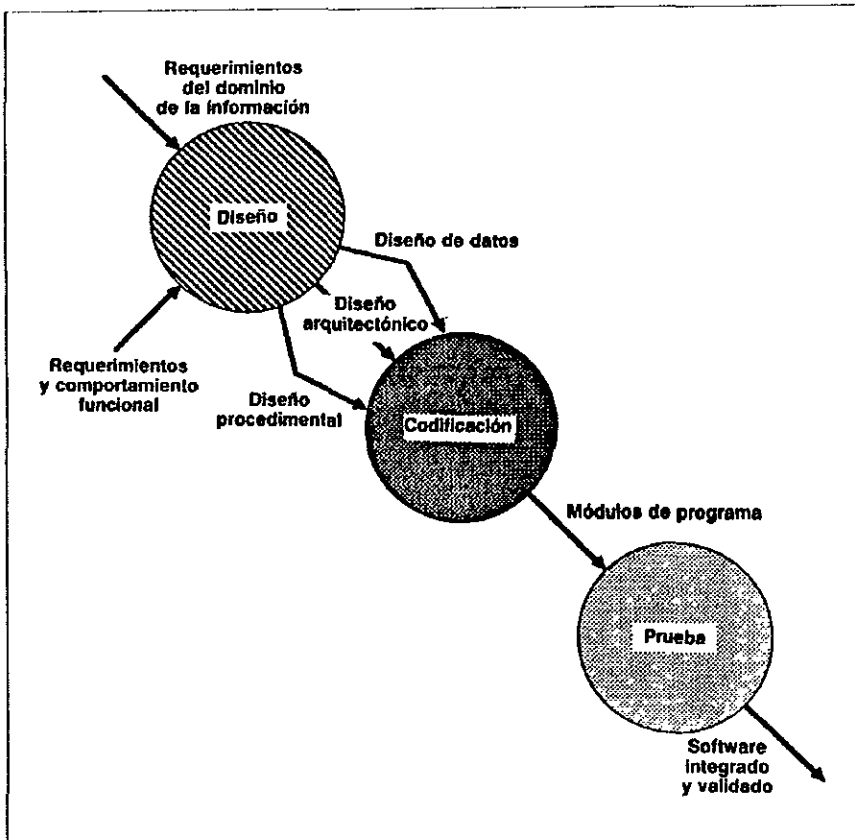


Figura 7.1. La fase de desarrollo.

Los requerimientos del programa, manifestados mediante el dominio de la información, requerimientos funcionales y de comportamiento, alimentan el paso del diseño. Usando una de las distintas metodologías de diseño existentes, se realiza el diseño de datos, el diseño arquitectónico y el diseño procedimental. El diseño de datos se enfoca sobre la definición de la estructura de los datos. El diseño arquitectónico define las relaciones entre los principales elementos estructurales del programa. El diseño procedimental transforma los elementos estructurales en una descripción procedimental del software. Para integrar y validar el software se genera y prueba el código fuente.

## 7.1 Diseño del software.

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. Puede ser definido como *el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física.*

Muchas metodologías de diseño se han desarrollado y aplicado en la industria. Cada una de ellas introduce heurísticas y notaciones propias, así como una visión algo restringida de lo que caracteriza a la calidad del diseño, pero todas tienen varias características comunes: 1) un mecanismo para la traslación del dominio de la información en representación de diseño; 2) una notación para representar los componentes funcionales y sus interfaces; 3) heurísticas para el refinamiento y partición y 4) criterios para la valoración de la calidad. Entre las metodologías más usuales se encuentran las siguientes:

- **Diseño orientado al flujo de datos.** Utiliza las características del flujo de la información para derivar la estructura del programa. Un diagrama de flujo de datos se convierte en una estructura de programa usando una o dos técnicas de análisis de diseño –análisis de transformación o de transacción.
- **Diseño orientado a la estructura de datos.** Se enfoca hacia el dominio de la información. Sin embargo, en vez de concentrarse sobre el flujo de datos, los métodos orientados a la estructura de datos utilizan la estructura de la información como el conductor de la derivación del diseño.
- **Diseño orientado al objeto.** Crea una representación del dominio del problema en el mundo real y lo transforma en un dominio de solución dentro del software. Esto da como resultado un diseño que interconexiona los objetos de datos (elementos de datos) y las operaciones de procesamiento, de forma que modulariza la información y el procesamiento en vez de sólo el procesamiento.

Cada metodología de diseño de software tiene sus puntos fuertes y débiles. Un factor importante de selección de un método de diseño es la amplitud de áreas a las que puede aplicarse. El método de diseño orientado al flujo de datos es particularmente útil cuando la información se procesa secuencialmente. Por ejemplo, procedimientos de análisis numérico, control de procesos y muchas otras aplicaciones de ingeniería y científicas caen dentro de esta categoría. Este enfoque es el que será utilizado para el desarrollo del programa de conversión de formatos.

## 7.1.1 Diseño arquitectónico y de datos.

### Consideraciones sobre el proceso de diseño orientado al flujo de datos.

El diseño orientado al flujo de datos permite una sencilla transición de las representaciones de la información (el diagrama de flujo de datos contenido en el Análisis de Requerimientos del Software) a una descripción de diseño de la estructura del programa. La transición desde el flujo de información a la estructura, se realiza como parte de un proceso de cinco pasos:

- 1) Se revisa el modelo del sistema fundamental (nivel 0 del Diagrama de Flujo de Datos).
- 2) Se refinan las burbujas del DFD correspondientes al software que se va a desarrollar.
- 3) Se establece el tipo de flujo de la información.
- 4) Se indican los límites del flujo.
- 5) El DFD se convierte en la estructura del programa.

El tipo de flujo de información (paso 3) es el conductor del método de conversión que se sigue en el paso 5 para obtener el diagrama de estructura del programa. En los siguientes párrafos se examinarán brevemente los dos tipos de flujo que se pueden presentar al efectuar el refinamiento del Diagrama de Flujo de Datos.

#### Flujo de transformación.

Recordando el modelo de sistema fundamental (nivel 0 del Diagrama de Flujo de Datos), la información debe entrar y salir en una forma del "mundo externo". Por ejemplo, los datos escritos sobre un teclado de una terminal, hablados sobre una línea de teléfono o dibujados sobre una presentación gráfica por computadora, son todas formas de información del mundo externo. Tales datos externos deben ser convertidos en una forma interna para su procesamiento.

La historia del tiempo del flujo de datos queda ilustrada por la figura 7.2. la información entra al sistema mediante caminos que transforman los datos externos en una forma interna y se identifica como flujo de entrada. En el núcleo del software ocurre una transición. Los datos de llegada se pasan a través del centro de transformación y comienzan a moverse a lo largo de caminos que conducen ahora hacia la "salida" del software. Los datos que se mueven por esos caminos se llaman flujo de salida. Cuando un segmento de un diagrama de flujo de datos exhibe estas características, se presenta un flujo de transformación.

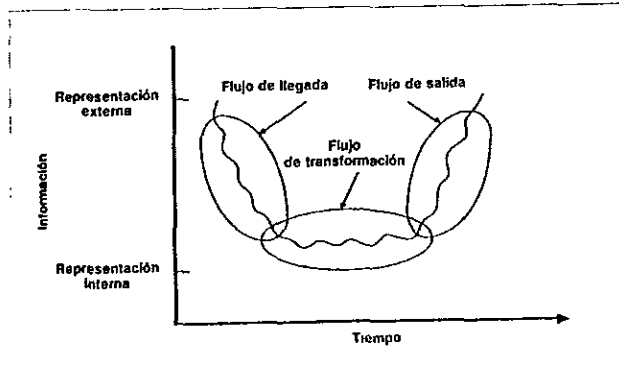


Figura 7.2. Flujo de transformación.

**Flujo de transacción.**

El modelo del sistema fundamental implica un flujo de transformación; por lo tanto, todo el flujo de datos puede colocarse en esta categoría. Sin embargo, el flujo de información se caracteriza frecuentemente por un elemento de datos sencillo, llamado una *transacción*, que desencadena otro flujo de datos a lo largo de uno de los *muchos caminos*. Cuando un DFD toma la forma mostrada en la figura 7.3, se presenta un *flujo de transacción*.

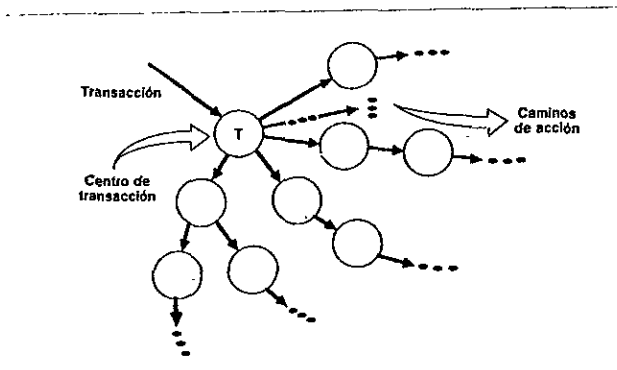


Figura 7.3. Flujo de transacción.

El flujo de transacción se caracteriza por datos que se mueven a lo largo de un camino de llegada que convierte la información del mundo externo en una transacción. La transacción es evaluada, y basándose en su valor, el flujo se inicia por uno de los muchos caminos de acción. El centro del flujo de información desde el que emanan muchos caminos de acción se llama un *centro de transacción*.



Resumen del proceso.

El método global de diseño orientado al flujo de datos se ilustra en la figura 7.4.

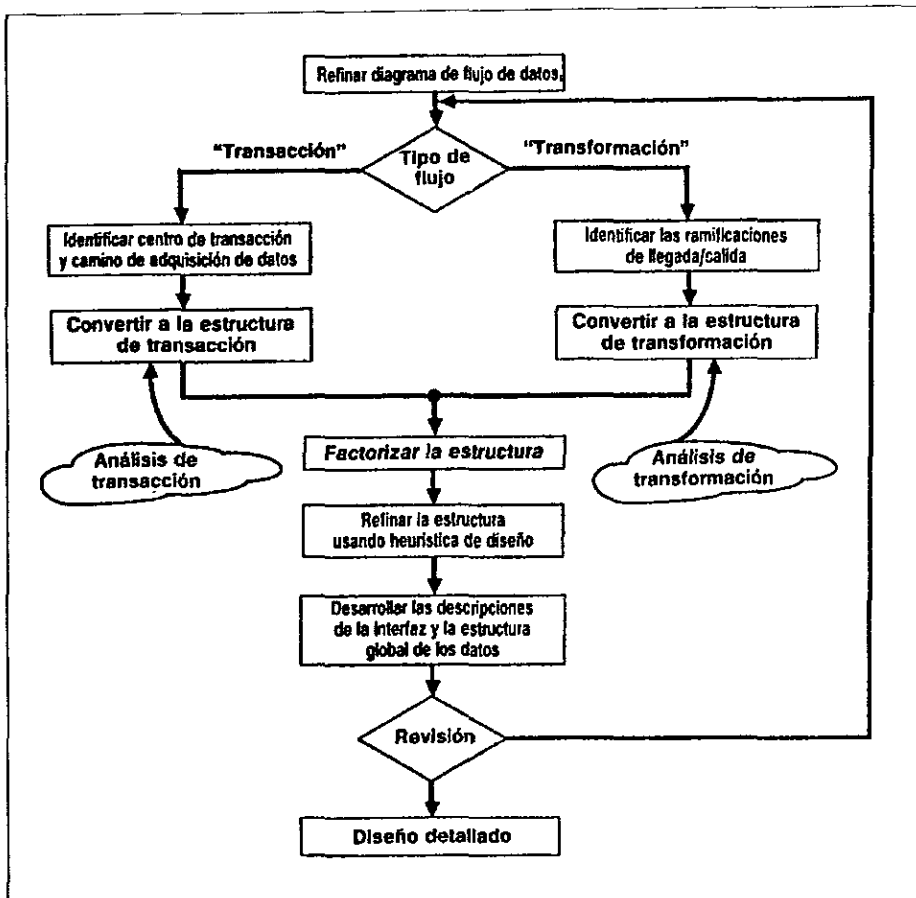


Figura 7.4. Diseño orientado al flujo de datos.

El diseño comienza con una evaluación de un diagrama detallado del flujo de datos. Se establece la categoría del flujo de la información (es decir, flujo de transformación o transacción) y se definen los límites del flujo que delinean el centro de transformación o transacción. Basándose en la posición de los límites, las transformaciones (las "burbujas" del DFD) se convierten en estructura de programa como módulos. La organización y definición precisa de los módulos se realiza mediante la distribución descendente de criterios para conseguir una modularidad efectiva.

## Pasos del diseño.

Los conceptos anteriores se usarán para ilustrar cada paso del diseño dentro del programa de conversión de formatos. Los pasos comienzan con una reevaluación del trabajo hecho durante la etapa de Análisis de Requerimientos del Software (sección 6.3) y concluyen con el desarrollo de la estructura del programa.

### Paso 1. Revisión del modelo de sistema fundamental.

El modelo de sistema fundamental abarca el nivel 0 del diagrama de flujo de datos y el soporte de la información. En realidad, el paso de diseño comienza con una evaluación de la especificación del sistema (sección 6.1) y del Análisis de los Requerimientos del Software (sección 6.3). Ambos documentos describen el flujo y estructura de la información de la interface del software. Las figuras de las páginas 104 y 105 del capítulo 6 describen el DFD para el programa de conversión.

### Paso 2. Revisión y refinamiento de los diagramas de flujo de datos para el software.

Usando la información obtenida en el Análisis de los Requerimientos del Software, se deriva un diagrama de flujo de datos a nivel 03 exclusivamente para la parte del proceso de conversión de formato HSR-11B a CGM, esto se ilustra en la página siguiente.

### Paso 3. Determinar si el DFD tiene características de transformación o de transacción.

Refiriéndonos al DFD de nivel 03, se observa que tiene unas características clásicas de flujo de transacción. Sin embargo, el flujo a lo largo de cada uno de los caminos de transformación que emana de la burbuja de llamada a las funciones del HSR-11B, aparece con características de flujo de transformación. Por lo tanto, deben establecerse límites para ambos tipos de flujo. En el mismo diagrama se observa que la información gráfica de salida sólo se produce en cuatro de los doce procesos que operan sobre los comandos del HSR-11B. Esto se debe a que el flujo de información gráfica de las secciones sísmicas, tal y como es entregada por el equipo rasterizador, únicamente hace uso de los comandos *Sectored Vectors* y *Seismic Traces* para codificar la sección.

Los procesos *Initialize* y *Graphic Flush* se utilizan al principio y al final del flujo de datos respectivamente, para dar información referente a la escala utilizada, las dimensiones de la gráfica y otros parámetros específicos para el graficador. Todos los restantes procesos en el DFD que no tienen flecha de salida no serán utilizados por el programa de conversión de formatos.

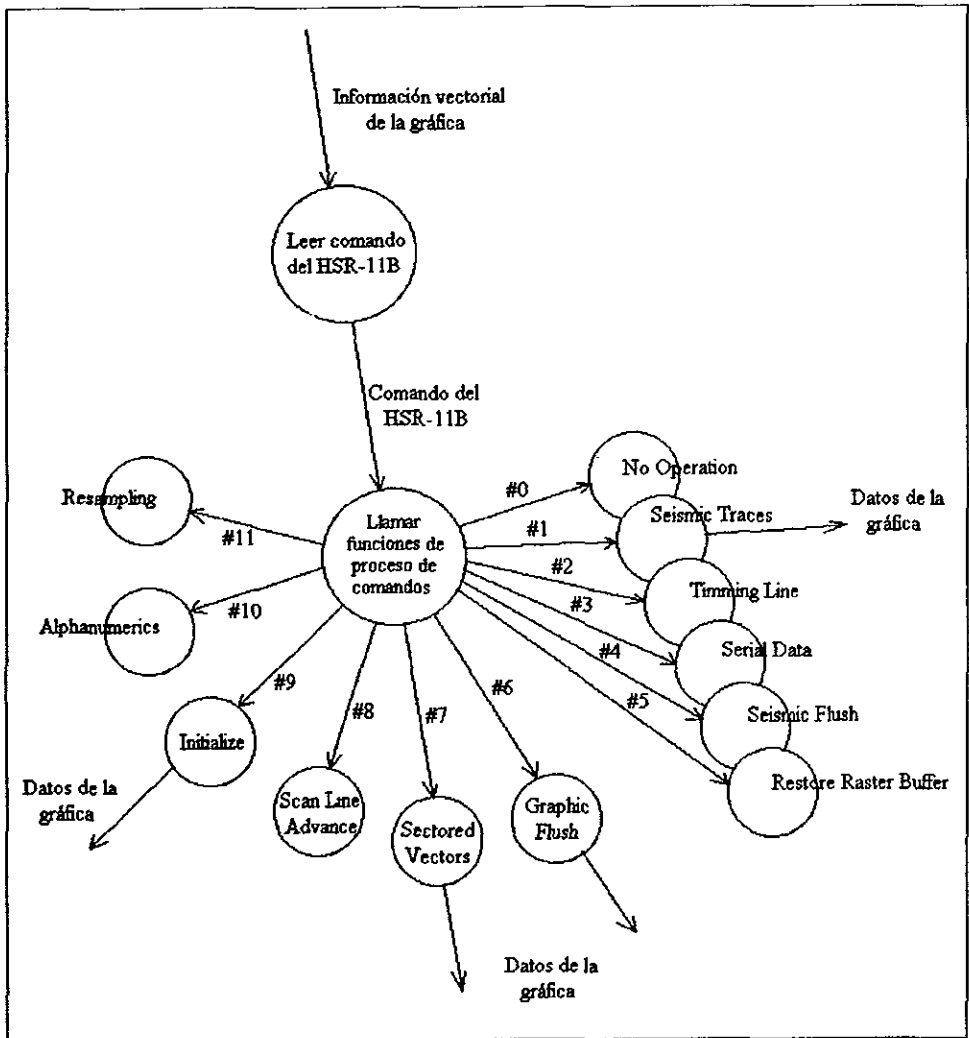


Figura 7.5. Refinamiento del proceso de conversión HSR-11B a CGM DFD de nivel 03.

**Paso 4. Identificar el centro de transacción y las características del flujo de cada camino de acción.**

La posición del centro de transacción puede discernirse inmediatamente a partir del DFD. El centro de transacción está ligado al origen de varios caminos de información que fluyen radialmente de él. En el caso del diagrama de flujo del programa de conversión de formatos, la burbuja de llamada a las funciones del HSR-11B es el centro de transacción.

El camino de llegada, es decir, la vía a lo largo del flujo que es recibida por una transacción junto con todos sus caminos de acción deben ser aislados. Los límites que definen un camino de recepción (llegada) y los cuatro caminos de acción posibles para el DFD de nivel 03 se muestran en las siguientes figuras. Cada camino de acción debe evaluarse en función de sus características individuales de flujo.

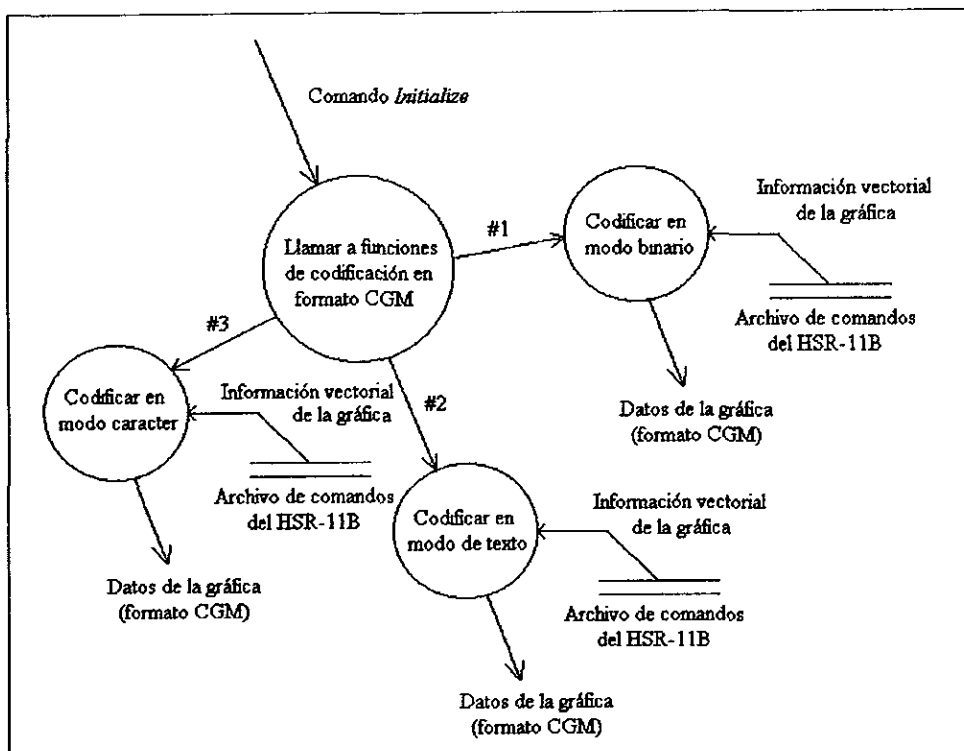


Figura 7.6. Aislamiento de los regímenes del flujo para el proceso *Initialize*. DFD de nivel 04

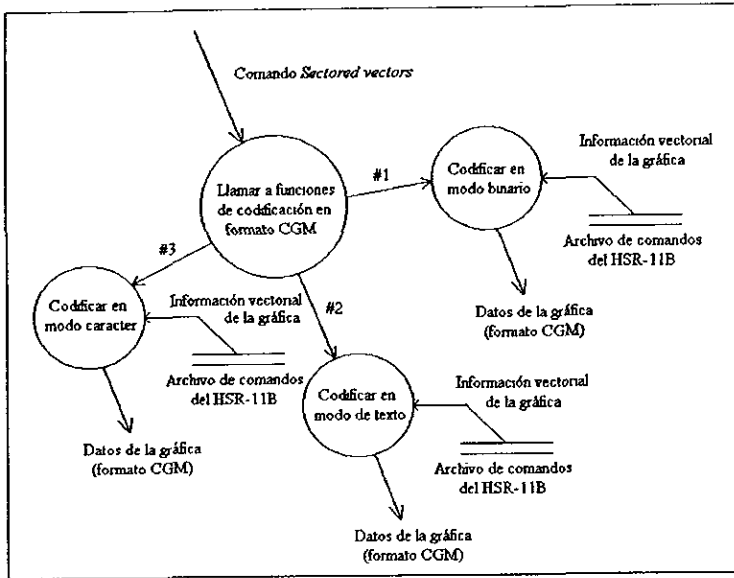


Figura 7.7. Aislamiento de los regimenes del flujo para el proceso *Sectored vectors*. DFD de nivel 04 (continuación)

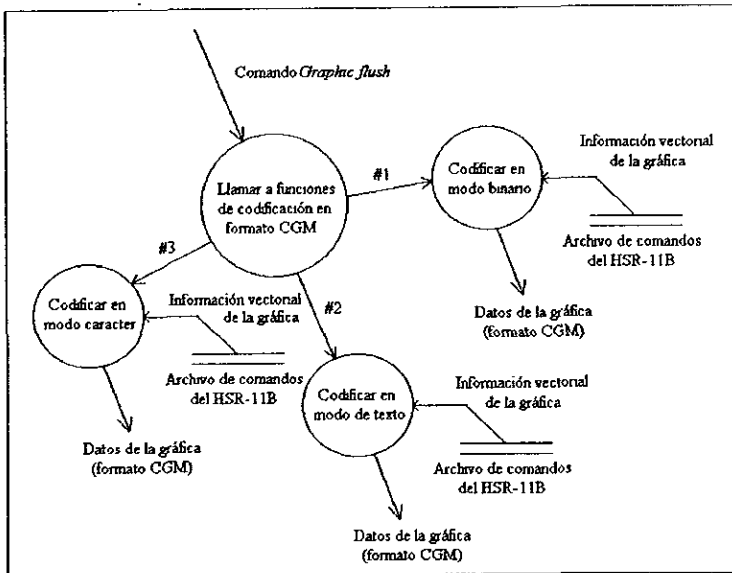


Figura 7.8 Aislamiento de los regimenes del flujo para el proceso *Graphic flush*. DFD de nivel 04 (continuación)

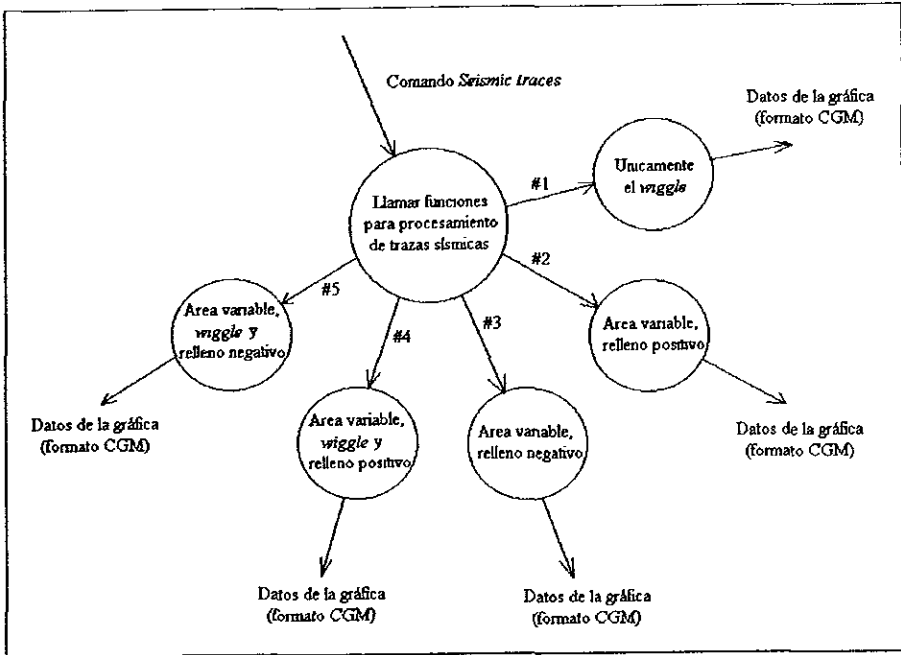


Figura 7.9 Aislamiento de los regímenes del flujo para el proceso *Seismic Traces*. DFD de nivel 04 (continuación)

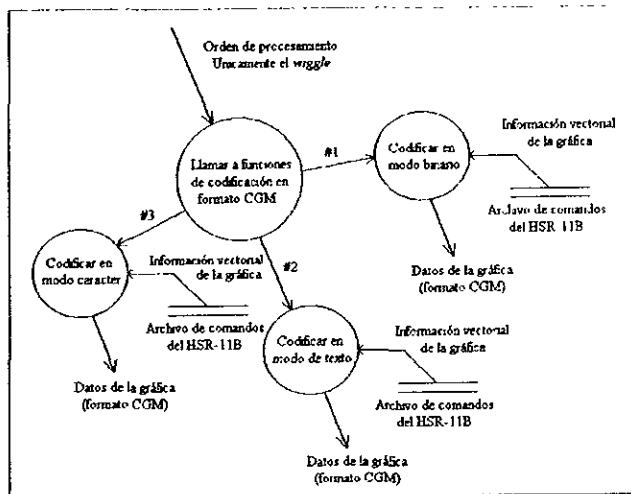


Figura 7.10. Aislamiento de los regímenes del flujo para el proceso *Unicamente el wiggle*. DFD de nivel 05

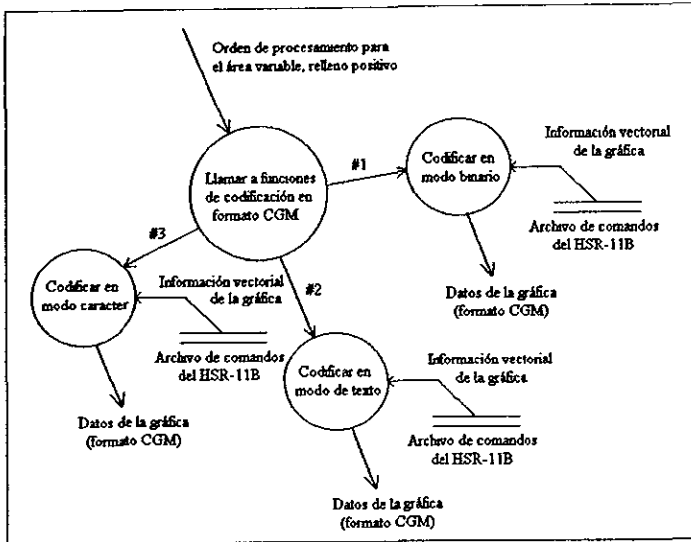


Figura 7.11. Aislamiento de los regímenes del flujo para el proceso *Area variable, relleno positivo*. DFD de nivel 05 (continuación).

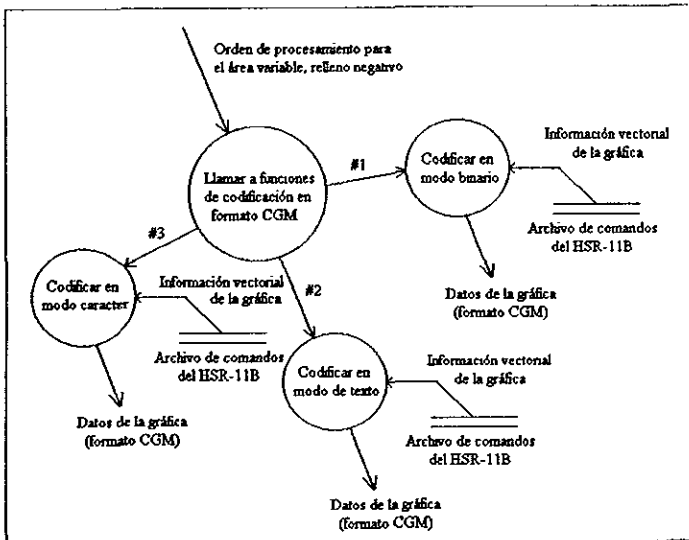


Figura 7.12. Aislamiento de los regímenes del flujo para el proceso *Area variable, relleno negativo*. DFD de nivel 05 (continuación).

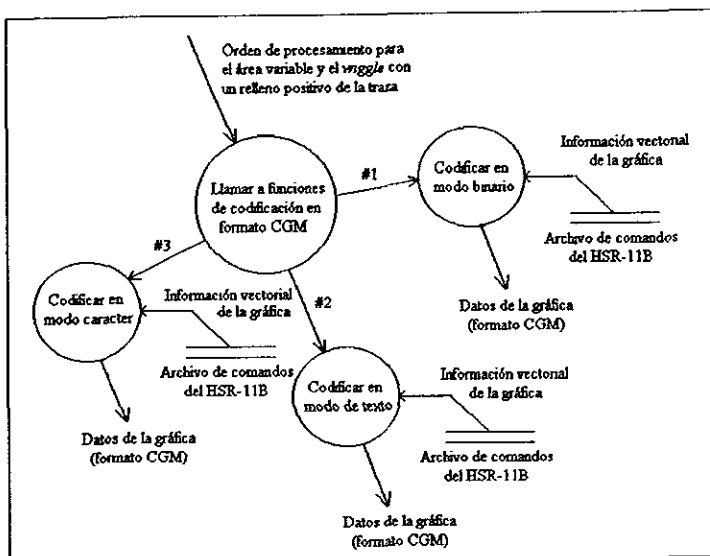


Figura 7.13. Aislamiento de los regímenes del flujo para el proceso *Área variable, wiggle y relleno positivo*. DFD de nivel 05 (continuación).

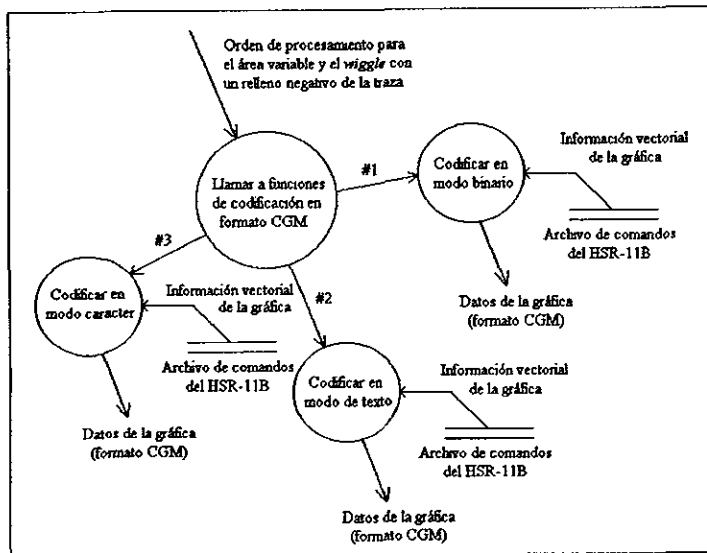


Figura 7.14. Aislamiento de los regímenes del flujo para el proceso *Área variable, wiggle y relleno negativo*. DFD de nivel 05 (continuación).



**Paso 5. Transformar el DFD en una estructura de software adecuada al procesamiento de transacciones.**

El flujo de transacción se convierte en una estructura de programa que contiene una bifurcación de entrada y una bifurcación de salida o expedición. La estructura para la bifurcación de entrada se desarrolla de la misma forma que en el análisis de transformaciones. Comenzando en el centro de la transacción, las burbujas a lo largo del camino de llegada se convierten en módulos. La estructura de la rama de expedición contiene un módulo expeditor que controla todos los módulos de acción subordinados. Cada camino del flujo de acción del DFD se convierte en una estructura que se corresponde con las características específicas del flujo. Este proceso se ilustra en la figura 7.15 para el programa de conversión de formatos. Aquí se observa que para las cuatro funciones que van a ser procesadas del equipo rasterizador la estructura es básicamente la misma. Los detalles de procesamiento para cada módulo se refinan con más detalle en la parte del diseño procedimental.

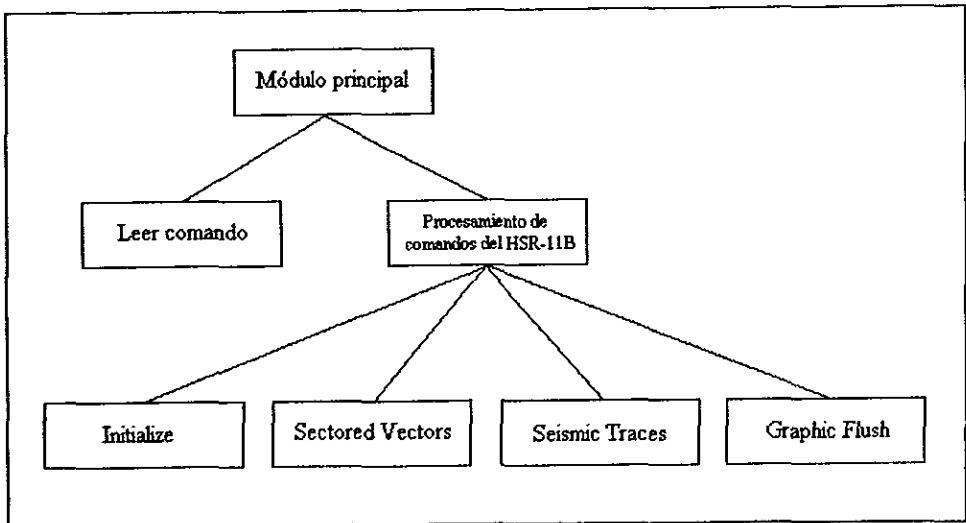


Figura 7.15. Conversión de transacciones para el diagrama de flujo de datos de nivel 03

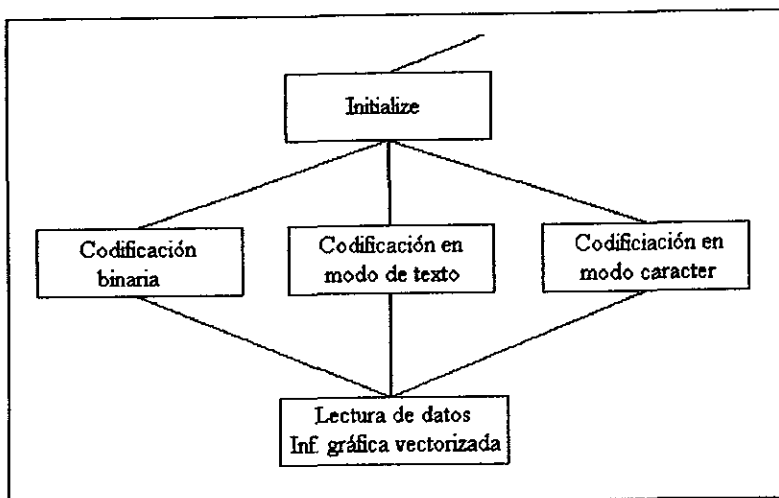


Figura 7.16. Desarrollo de la estructura del camino de acción para el módulo Initialize.

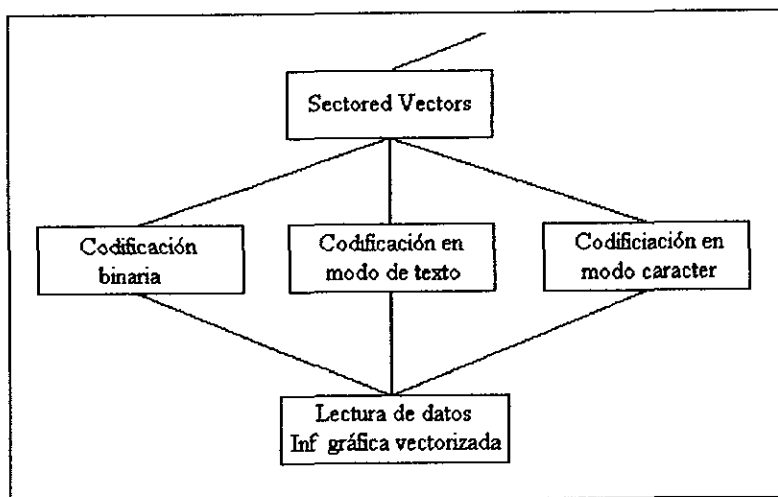


Figura 7.17. Desarrollo de la estructura del camino de acción para el módulo Sectored Vectors.

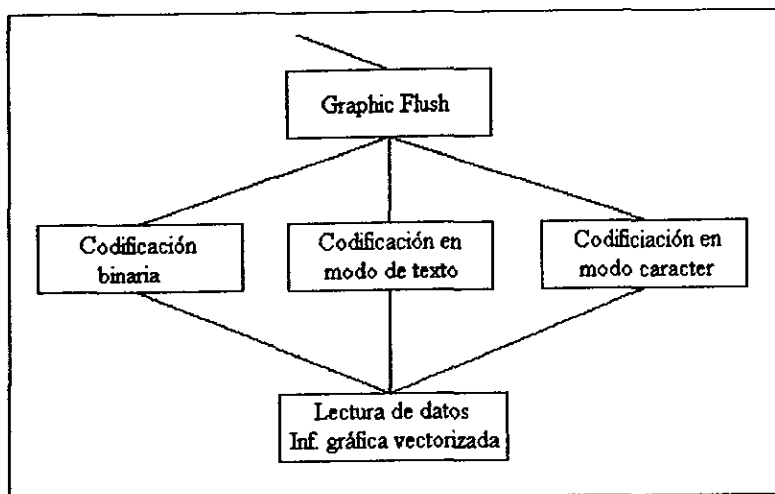


Figura 7.18. Desarrollo de la estructura del camino de acción para el módulo Graphic Flush.

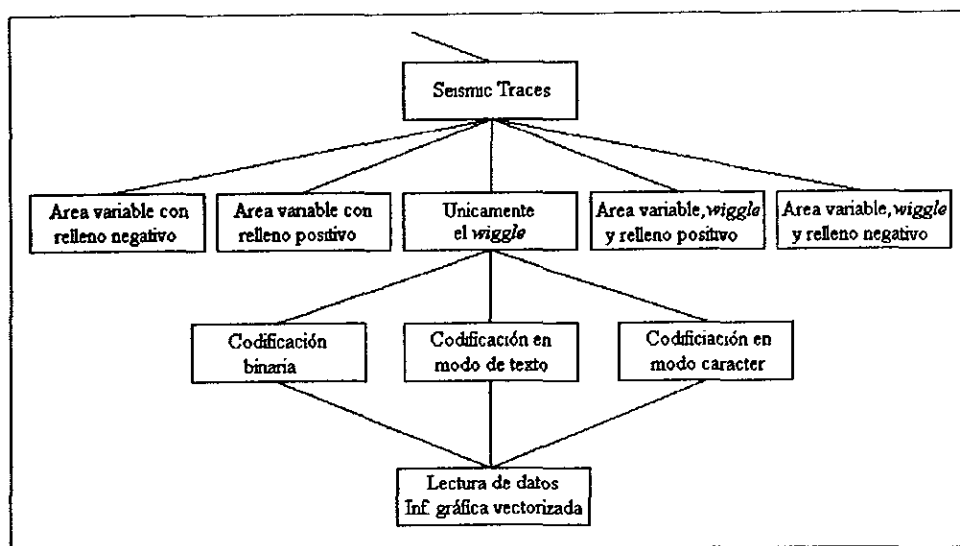


Figura 7.19. Desarrollo de la estructura del camino de acción para el módulo Seismic Traces. Los módulos de codificación binaria, en modo de texto y en modo carácter que aquí sólo se muestran para un módulo se repiten para todos los restantes.

### 7.1.2 Diseño procedimental.

Una vez que ha sido establecida la estructura del programa se realiza el diseño procedimental definiendo sin ambigüedad los detalles algorítmicos que deben establecerse. Para ello se utilizan diversas representaciones –algunas gráficas, otras tabulares o basadas en lenguajes, de las cuales la técnica del diagrama de flujos es la más ampliamente utilizada.

Diagrama de flujo del módulo principal del programa de conversión

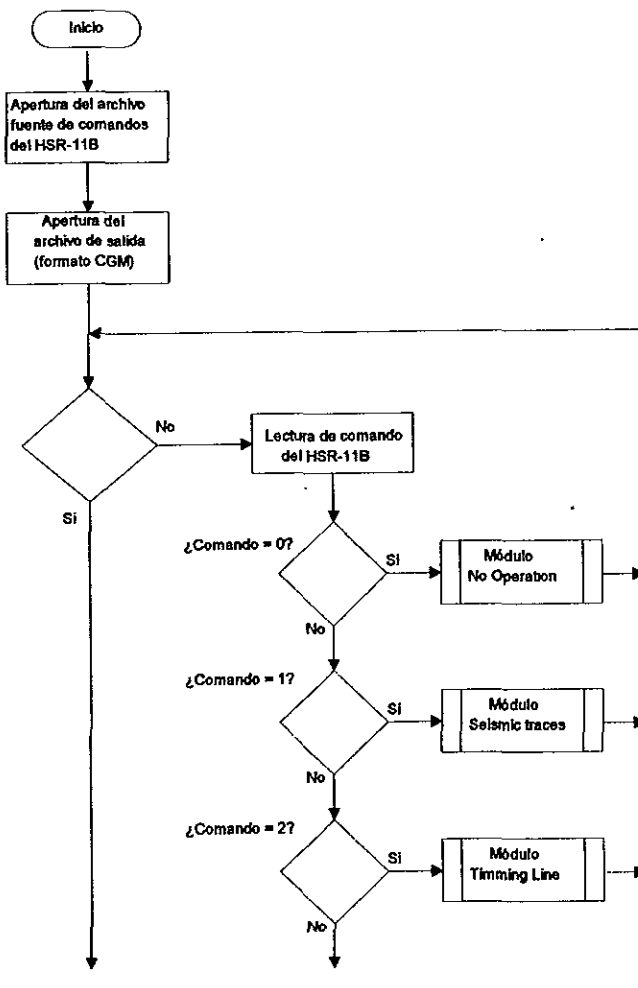


Diagrama de flujo del módulo principal del programa de conversión (continuación)

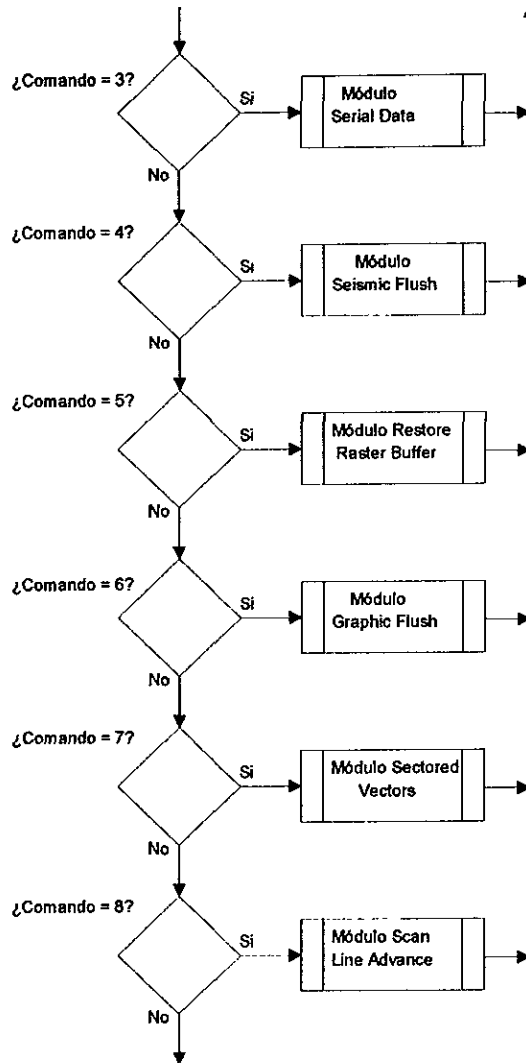


Diagrama de flujo del módulo principal del programa de conversión (continuación)

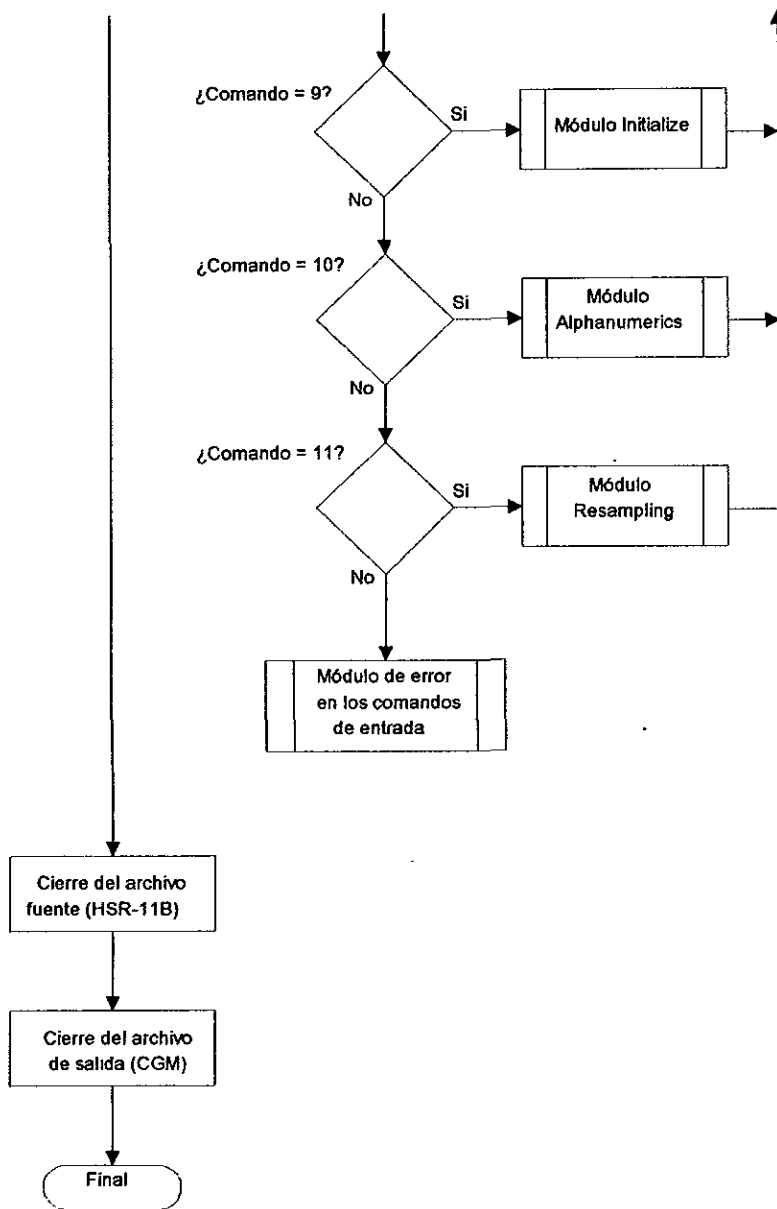
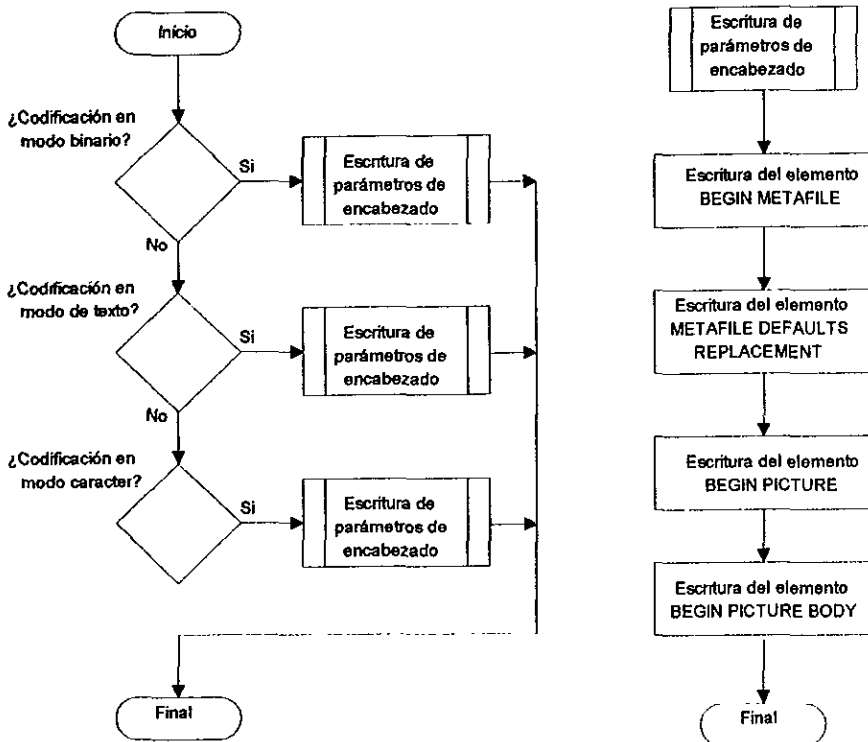
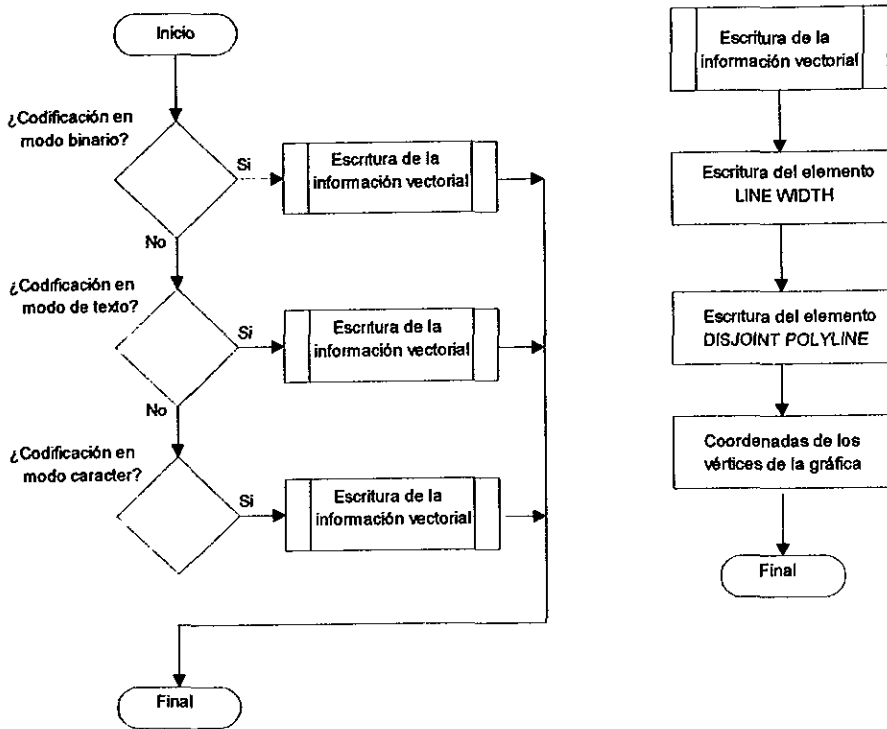


Diagrama de flujo del módulo Initialize



La subrutina de escritura de parámetros del encabezado sigue un orden secuencial que es idéntico para cada modalidad de codificación, la diferencia radica en la forma que adoptan los datos de salida en el archivo CGM generado. En el diagrama de la derecha se ilustra la secuencia de este procedimiento.

Diagrama de flujo del módulo Sectored Vectors



En este otro diagrama se ilustra la secuencia de escritura para un bloque de información vectorial Sin importar la modalidad de codificación, el proceso es el mismo, diferenciándose únicamente en la estructura de los datos de salida. En primer lugar se codifica el elemento *LINEWIDTH*, que establece el grosor de las líneas. El elemento *DISJOINT POLYLINE* sirve para codificar vectores individuales. Por último, las coordenadas de los vértices se refieren a un conjunto de parejas de valores que constituyen los extremos de cada uno de los vectores que dan forma a la gráfica.



Diagrama de flujo del módulo Seismic Traces

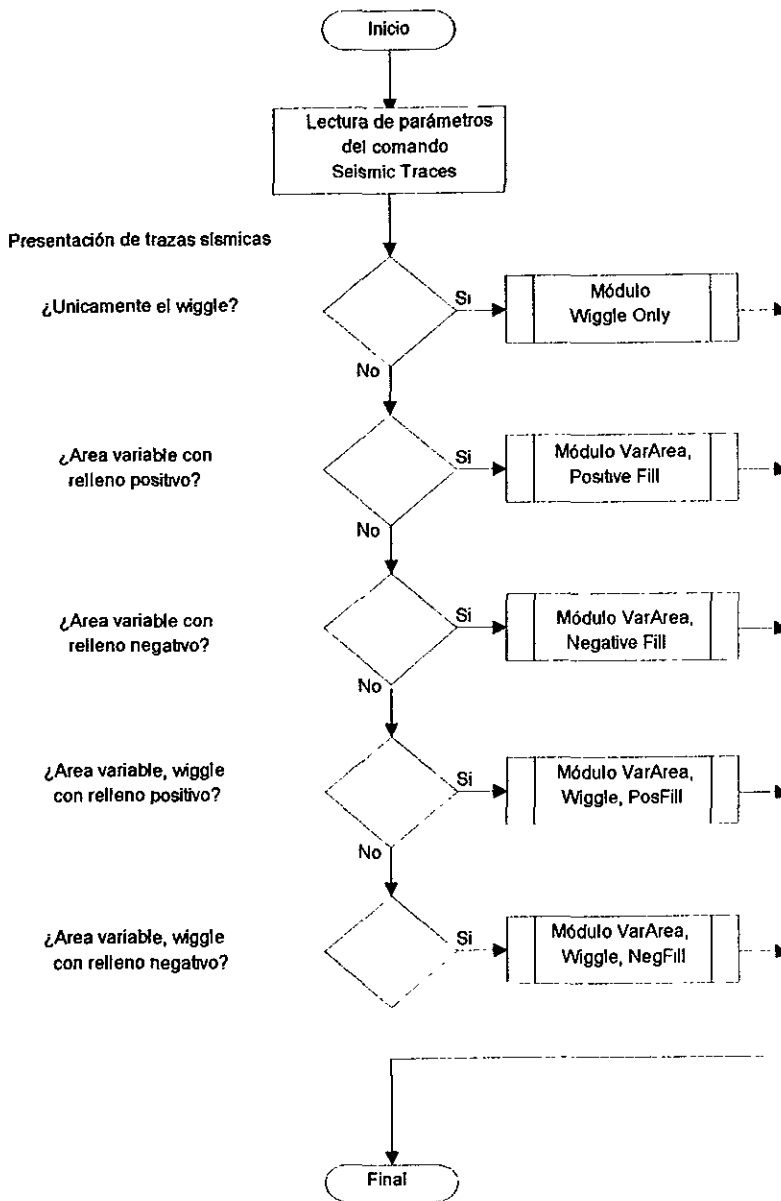
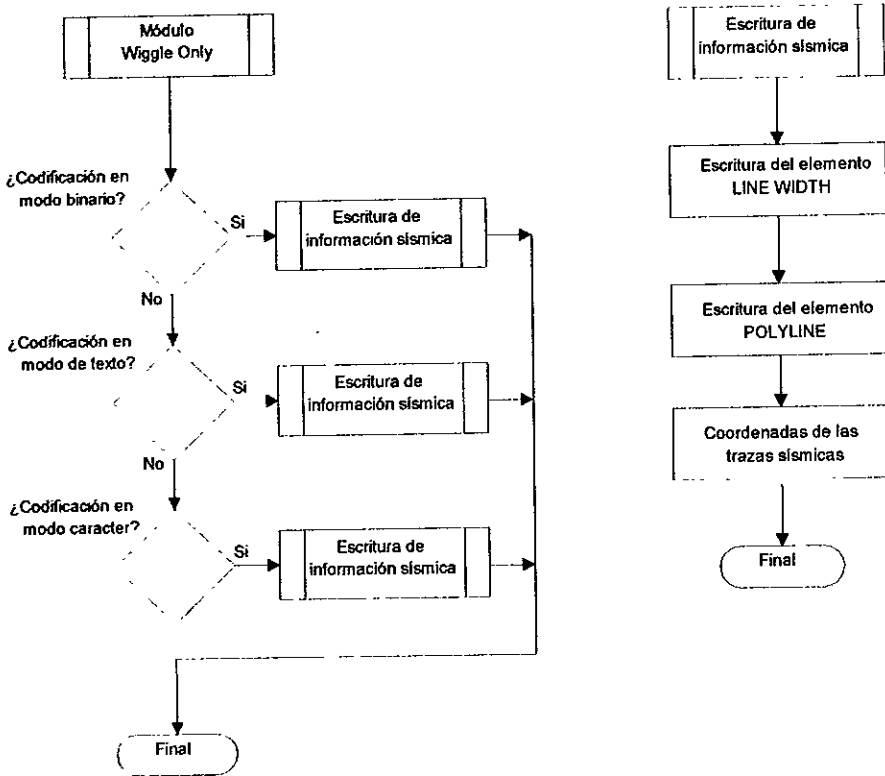


Diagrama de flujo del módulo Seismic Traces (continuación)



En la codificación de las trazas sísmicas considerando únicamente el *wiggle*, se utiliza el elemento gráfico POLYLINE, el cual traza una línea continua a través de un conjunto de puntos. La subrutina de escritura de información sísmica es esencialmente idéntica para las cinco modalidades de despliegue, diferenciándose solamente en el uso de los elementos POLYLINE, POLYGON o una combinación de ambos. Esto se ilustra en los siguientes diagramas de flujo para los procedimientos restantes del módulo Seismic Traces.

Diagrama de flujo del módulo Seismic Traces (continuación)

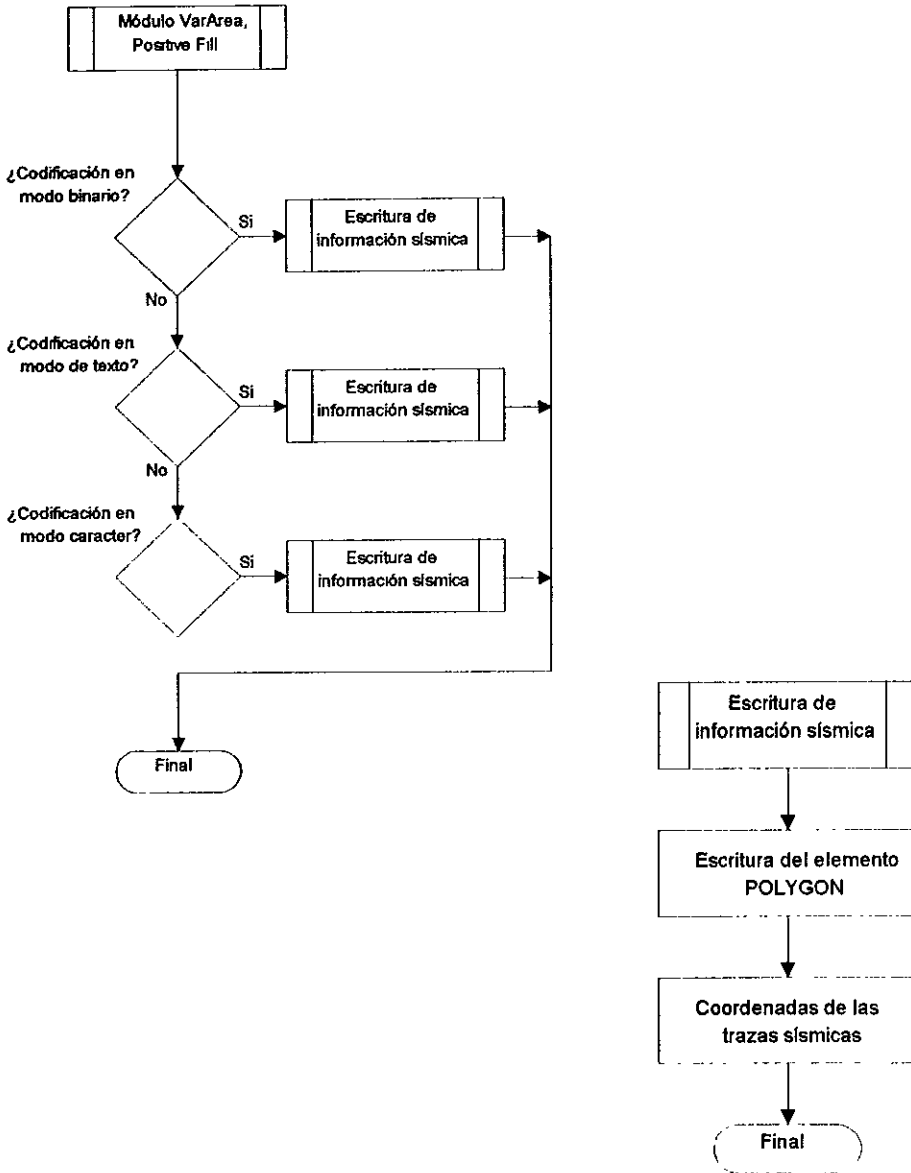


Diagrama de flujo del módulo Seismic Traces (continuación)

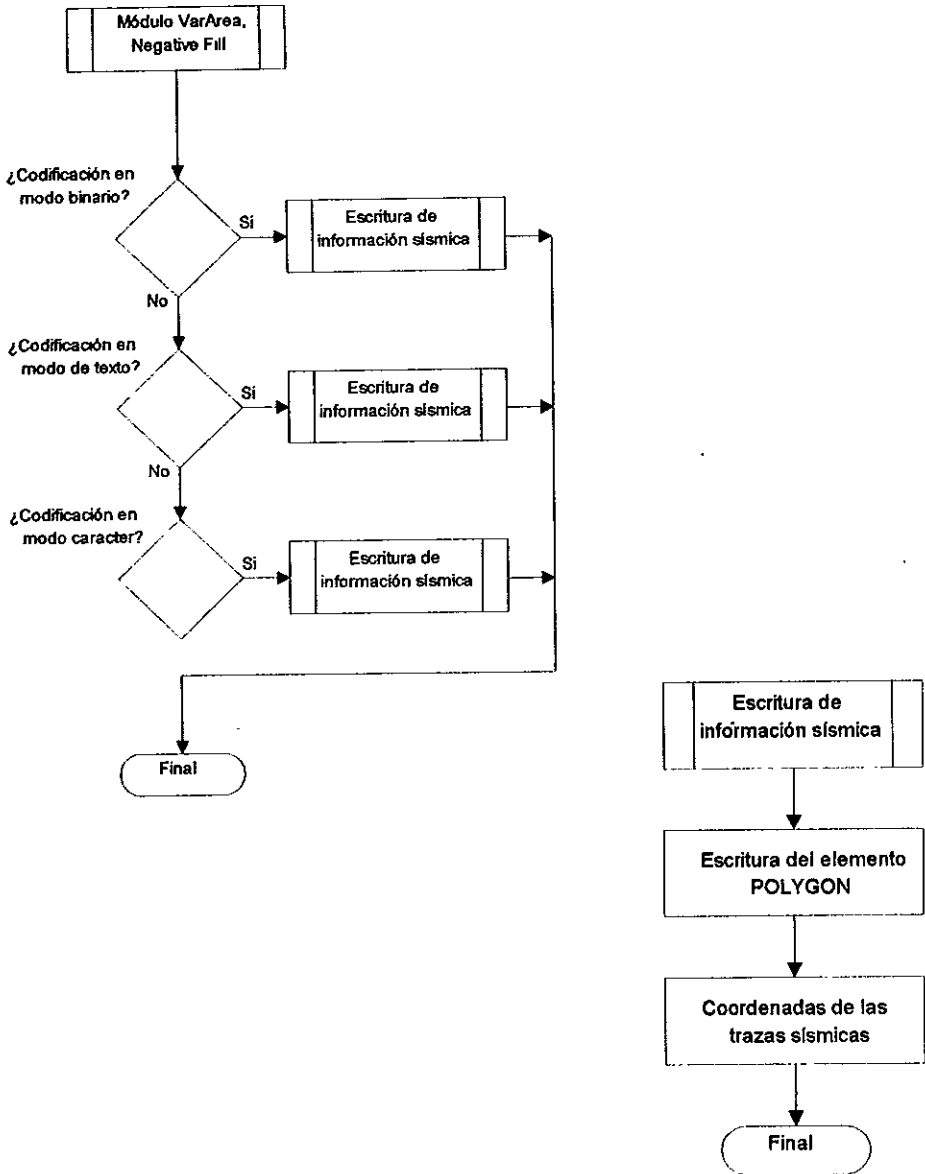


Diagrama de flujo del módulo Seismic Traces (continuación)

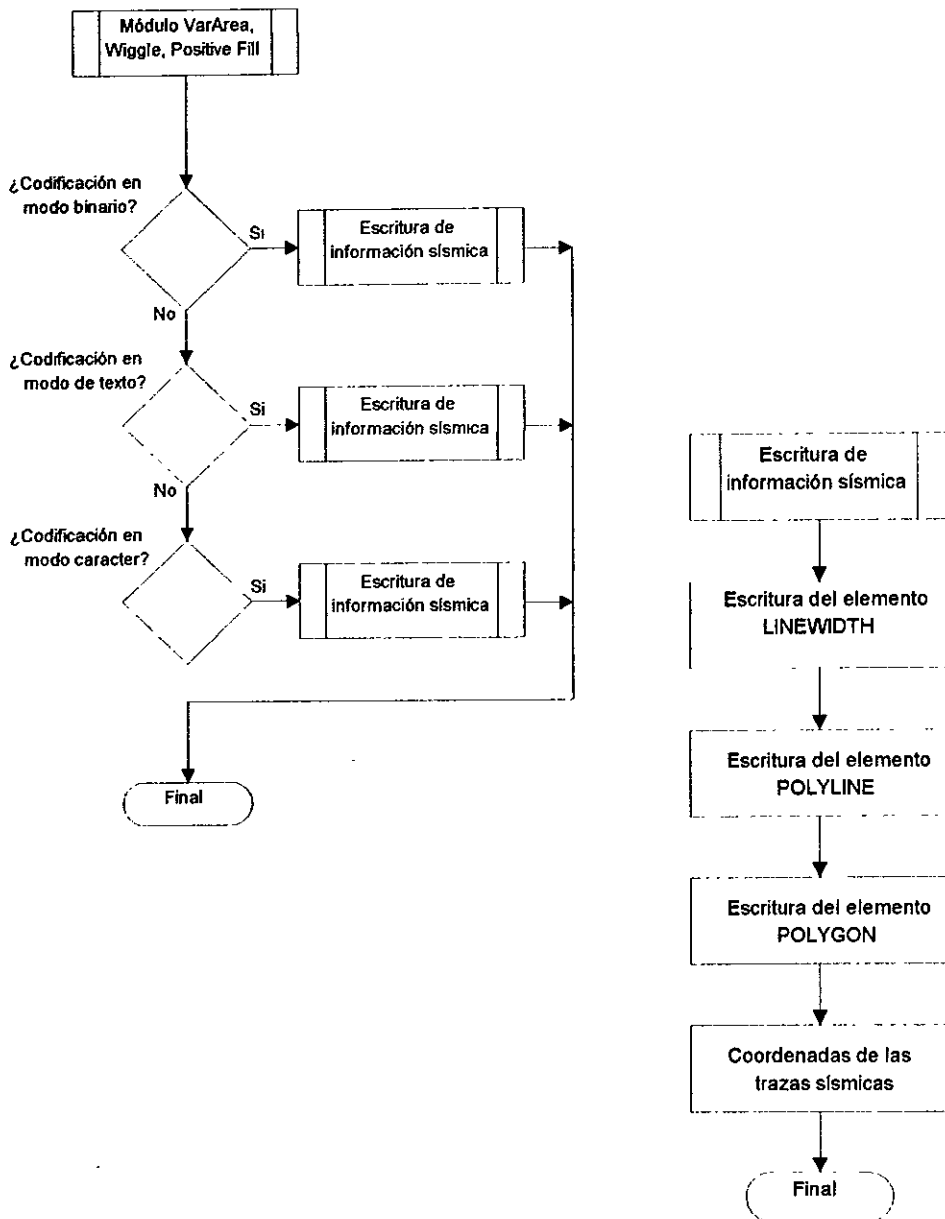
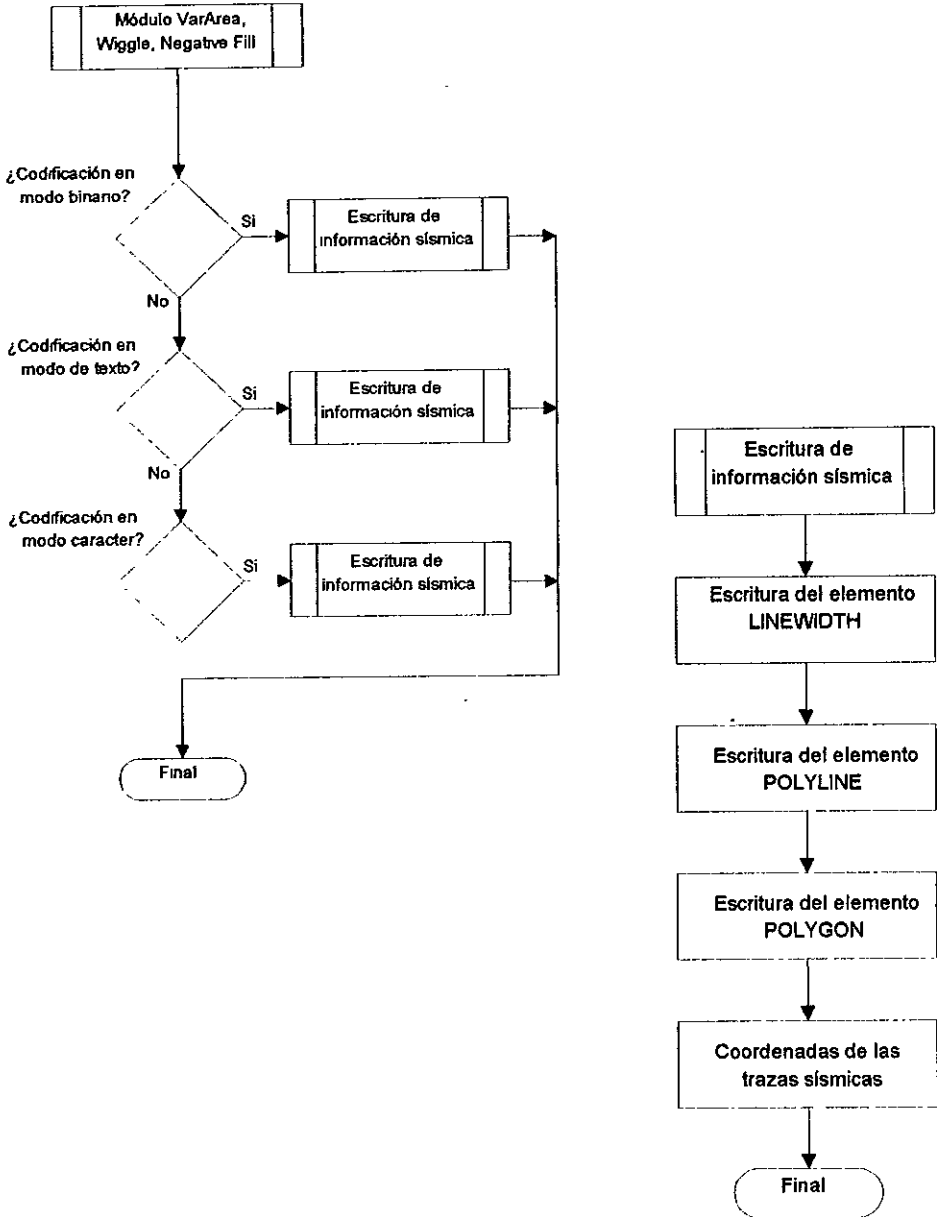


Diagrama de flujo del módulo Seismic Traces (continuación)



## 7.2 Codificación.

Todos los pasos de la ingeniería del software que se han presentado hasta ahora van dirigidos hacia un objetivo final: traducir las representaciones del software a una forma que pueda ser “comprendida” por una computadora. El paso de la codificación traduce una representación del software dada por un diseño detallado a una realización en un lenguaje de programación. El proceso de traducción continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto dependiente de la máquina. Más tarde la salida del compilador es traducida a código máquina –las instrucciones reales que dirigen la lógica cableada o microprogramada de la unidad central de proceso.

Idealmente la elección del lenguaje de programación para la etapa de codificación debe tener en cuenta las características del problema planteado. Sin embargo, en el caso del programa de conversión de formatos, la selección del lenguaje de programación C obedeció a que en los ambientes UNIX es con frecuencia el único lenguaje utilizado para el desarrollo de aplicaciones. La versión utilizada fue en este caso el XLC Compiler versión 1.3.0.0 de IBM en el ambiente del sistema operativo AIX 4.2.

Cada uno de los módulos fuente del proyecto se codificaron como archivos independientes, los cuales fueron posteriormente compilados como un proyecto único. Debido a la imposibilidad de transcribir directamente el código fuente de todos los módulos en el presente trabajo dado su tamaño, en las siguientes páginas se muestran los listados de los módulos principales del programa, a saber:

- **hsr2cgm.c** Representa el módulo principal del programa. Se ocupa de procesar los comandos del archivo de entrada e invocar los procedimientos requeridos conforme a los diferentes comandos del sistema HSR-11B.
- **sectordv.c** Módulo secundario encargado de codificar toda la información vectorial de la gráfica.
- **seismict.c** Módulo secundario que se encarga de procesar la información sísmica.

```

/* HSR2CGM.C, modulo principal del programa de conversion */
/* */
/* Proposito: generar un archivo en formato de CGM (Computer Graphics
/* Metafile) a partir de informacion grafica en forma de comandos
/* nativos del sistema HSR-11B (High Speed Rasterizer). */
/* */
/* Ejemplo de llamada: hsr2cgm [-b|-t|-c] fuente[.hsrc] [destino] [presentacion]
/* */
/* Entradas: desde la linea de ordenes del sistema operativo, el tipo de archivo
/* en formato CGM que se desea generar, el archivo de comandos de
/* entrada, el nombre del archivo de salida y el modo de despliegue de las
/* trazas sismicas. */
/* */
/* Salidas: archivo grafico en formato cgm con las características
/* especificadas en la linea de entrada */
/* */
/* Subrutinas referenciadas: NoOperation(), SeismicTraces(), TimingLine(),
/* SerialData(), SeismicFlush(),
/* RestoreRasterBuffer(), GraphicFlush(),
/* SectoredVectors(), ScanLineAdvance(),
/* Initialize(), Alphanumeric(), Resampling()
/* */
/* Datos pertinentes: modulo desarrollado en lenguaje de programacion C,
/* apegado al estandar ANSI-C 1989 para ser ejecutado en
/* ambiente UNIX asi como de MS-DOS. Este programa forma
/* parte de un proyecto de investigacion para la Gerencia
/* de Prospeccion Geofisica del Instituto Mexicano del Petroleo (IMP).
/* */
/* Autor: Miguel Angel Avila Martinez */
/* */
/* Fecha: 5/Junio/1998 */
/* */
/* Modificaciones: */
/* */
/* */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "hsr2cgm.h"

void main(int argc, char *argv[]) {

    FILE *fp1, *fp2; /* fp1 y fp2 son apuntadores a los archivos de entrada */
                    /* salida respectivamente */

    unsigned short int command, linethick, l, i;
    char inputfile[256], outputfile[256], mode[256], presentation[256];
    unsigned char byte, byte2;

    /* La siguiente estructura switch-case verifica los parametros de entrada que se le pasan */
    /* al programa a partir del prompt del sistema operativo. */

    switch(argc) {
    case 1: printf("hsr2cgm: programa de conversion de archivos graficos ");
            printf("en comandos nativos del\n");
            printf("sistema HSR-11B a formato cgm.\n");
            printf("\n");
            printf("Utllice: hsr2cgm [-b|-t|-c] fuente[ hsrc] [destino] [presentacion\n");
            printf("\n");
            printf("Donde\n");
            printf(" -b: conversion a formato binario (opcion por defecto).\n");
            printf(" -c: conversion a formato en modo caracter.\n");
            printf(" -t: conversion a formato en modo texto.\n");
            printf("\n");
            printf("Si no se proporciona el nombre del archivo destino, este tendra el ");
            printf("mismo nombre\n");
            printf("que el archivo fuente pero con la extension correspondiente al tipo ");
            printf("de grafica\n");
    }
}

```



```

printf("que se desee generar.\n");
printf("\n");
printf(" .cgm para archivos binarios.\n");
printf("  ctk para archivos en modo de texto.\n");
printf("  chr para archivos en modo caracter \n");
printf("\n");
printf(" El parametro opcional 'presentacion' permite forzar el modo de ");
printf("despliegue de\n");
printf("la información sísmica en el archivo gráfico de salida, de acuerdo ");
printf("a una de las\n");
printf("siguientes opciones:\n");
printf("\n");

#if PLATFORM
printf("- Ms -");
printf("\n");
printf("\n");
getch();
#else
printf("- Mas -");
printf("\n");
printf("\n");
getchar();
#endif

printf("Parametro 'presentacion'\nSignificado\n");
printf("\n");
printf(" -vaffi\W\WArea variable, relleno positivo\n");
printf(" -vanfi\W\WArea variable, relleno negativo\n");
printf(" -wiggie\W\WÚnicamente el wiggle\n");
printf(" -vawpfi\W\WArea variable, wiggle, relleno positivo\n");
printf(" -vawnfi\W\WArea variable, wiggle, relleno negativo\n");
printf("\n");
printf(" Si se omite este parametro las trazas sísmicas adoptaran la forma ");
printf("especificada\n");
printf("en el archivo de entrada de comandos del HSR-11B.\n");

#if PLATFORM
.
#else
printf("\n");
#endif

exit(1);

break;

case 2: strcpy(inputfile, argv[1]);
strcpy(mode, "-b");

if (strchr(inputfile, '.') != NULL) {
if (strstr(inputfile, ".HSR") == NULL && strstr(inputfile, ".HSr") == NULL &&
strstr(inputfile, ".HsR") == NULL && strstr(inputfile, ".hSP") == NULL &&
strstr(inputfile, ".Hsr") == NULL && strstr(inputfile, ".hSr") == NULL &&
strstr(inputfile, ".hsR") == NULL && strstr(inputfile, ".hsr") == NULL) {

printf("Error en la especificacion del archivo de entrada.\n");
printf("Útilice hsr2cgm [{-b|-l|-c}] fuente[ hsrj ] [destino] ").
printf("[presentacion]\n");
printf("Programa terminado.\n");

exit(1);
}
}

i = 0;

while(inputfile[i] != '\0') {
outputfile[i] = inputfile[i];
++i;
}

```

```

        outputfile[i] = '\0';
    } else {
        strcpy(outputfile, inputfile);
        strcat(inputfile, " hsr");
    }

    strcat(outputfile, ".cgm");

    break;

case 3. strcpy(mode, argv[1]);

if(mode[0] != '-') {
    strcpy(inputfile, argv[1]);
    strcpy(outputfile, argv[2]);
    strcpy(mode, "-b");

    if(!strcmp(outputfile, "-vapfil") || !strcmp(outputfile, "-vanfil") ||
        !strcmp(outputfile, "-wiggle") || !strcmp(outputfile, "-vawpfil") ||
        !strcmp(outputfile, "-vawnfil")) {

        strcpy(presentation, outputfile);
        j = strlen(inputfile);

        for(i = 0; i < j; ++i)
            if(inputfile[i] != '.')
                outputfile[i] = inputfile[i];

        outputfile[i] = '\0';
    }

    if (strchr(inputfile, '.') != NULL) {
        if (strstr(inputfile, ".HSR") == NULL && strstr(inputfile, ".hsr") == NULL &&
            strstr(inputfile, ".HsR") == NULL && strstr(inputfile, ".hSR") == NULL &&
            strstr(inputfile, ".Hsr") == NULL && strstr(inputfile, ".hSr") == NULL &&
            strstr(inputfile, ".hsR") == NULL && strstr(inputfile, ".hsr") == NULL) {

            printf("Error en la especificacion del archivo de entrada.\n");
            printf("Utllice: hsr2cgm [-b|-l|-c] fuente[ hsr] [destino] ");
            printf("[presentacion]\n");
            printf("Programa terminado.\n");

            exit(1);
        }
    }
} else
    strcat(inputfile, " hsr");

if(strchr(outputfile, '.') == NULL)
    strcat(outputfile, ".cgm");

} else {
    if((strcmp(mode, "-b") != 0) && (strcmp(mode, "-l") != 0) &&
        (strcmp(mode, "-c") != 0)) {

        printf("Error: argumento de conversion de formato no valido.\n");
        printf("Utllice: hsr2cgm [-b|-l|-c] fuente[ hsr] [destino] ");
        printf("[presentacion]\n");
        printf("Programa terminado.\n");

        exit(1);
    }

    strcpy(inputfile, argv[2]);

    if (strchr(inputfile, '.') != NULL) {
        if (strstr(inputfile, ".HSR") == NULL && strstr(inputfile, ".HSR") == NULL &&

```

```

    strstr(inputfile, ".HsR") == NULL && strstr(inputfile, ".hSP") == NULL &&
    strstr(inputfile, ".Hsr") == NULL && strstr(inputfile, ".hSr") == NULL &&
    strstr(inputfile, ".hsR") == NULL && strstr(inputfile, ".hsr") == NULL) {

    printf("Error en la especificacion del archivo de entrada.\n");
    printf("Utllice: hsr2cgm [{-b|-t|-c}] fuente[.hsr] [destino] ");
    printf("[presentacion]\n");
    printf("Programa terminado.\n");

    exit(1);
}

i = 0;

while(inputfile[i] != '\0') {
    outputfile[i] = inputfile[i],
    ++i;
}

outputfile[i] = '\0';

} else {
    strcpy(outputfile, inputfile);
    strcal(inputfile, ".hsr");
}

if(!strcmp(mode, "-b"))
    strcal(outputfile, "cgm");
else if(!strcmp(mode, "-t"))
    strcal(outputfile, "cli");
else if(!strcmp(mode, "-c"))
    strcal(outputfile, "chr");
}

break;

case 4: strcpy(mode, argv[1]);
        strcpy(inputfile, argv[2]);
        strcpy(outputfile, argv[3]);

        if(mode[0] == '-') {
            if((strcmp(mode, "-b") != 0) && (strcmp(mode, "-t") != 0) &&
                (strcmp(mode, "-c") != 0)) {

                printf("Error: argumento de conversion de formato no valido.\n");
                printf("Utllice: hsr2cgm [{-b|-t|-c}] fuente[.hsr] [destino] ");
                printf("[presentacion]\n");
                printf("Programa terminado.\n");

                exit(1);
            }

            if((!strcmp(outputfile, "-vapfil") || !strcmp(outputfile, "-vanfil") ||
                !strcmp(outputfile, "-wiggie") || !strcmp(outputfile, "-vawpfil") ||
                !strcmp(outputfile, "-vawnfil"))) {

                strcpy(presentation, outputfile);
                j = strlen(inputfile);

                for(i = 0; i < j; ++i)
                    if(inputfile[i] != '\0')
                        outputfile[i] = inputfile[i],

                outputfile[i] = '\0',
            }

        } else {
            strcpy(presentation, outputfile);
            strcpy(outputfile, inputfile);

```

```

strcpy(inputfile, mode);
strcpy(mode, "-b");

if((strcmp(presentation, "-vapfil") != 0) &&
    (strcmp(presentation, "-vanfil") != 0) &&
    (strcmp(presentation, "-wiggie") != 0) &&
    (strcmp(presentation, "-vawpfil") != 0) &&
    (strcmp(presentation, "-vawnfil"))) {

    printf("Error en el parametro de presentacion de la informacion sismica \n");
    printf("Utilice: hsr2cgm [-b|-l|-c] fuente[.hsr] [destino] ");
    printf("[presentacion]\n");
    printf("Programa terminado.\n");

    exit(1);
}
}

if (strchr(inputfile, '.') != NULL) {
    if (strstr(inputfile, ".HSR") == NULL && strstr(inputfile, ".HSr") == NULL &&
        strstr(inputfile, ".HsR") == NULL && strstr(inputfile, ".hSR") == NULL &&
        strstr(inputfile, ".Hsr") == NULL && strstr(inputfile, ".hSr") == NULL &&
        strstr(inputfile, ".hsR") == NULL && strstr(inputfile, ".hsr") == NULL) {

        printf("Error en la especificacion del archivo de entrada.\n");
        printf("Utilice: hsr2cgm [-b|-l|-c] fuente[.hsr] [destino] ");
        printf("[presentacion]\n");
        printf("Programa terminado.\n");

        exit(1);
    }
} else
    strcat(inputfile, ".hsr");

if (strchr(outputfile, '.') == NULL) {
    if(!strcmp(mode, "-b"))
        strcat(outputfile, ".cgm");
    else if(!strcmp(mode, "-l"))
        strcat(outputfile, ".cit");
    else if(!strcmp(mode, "-c"))
        strcat(outputfile, ".chr");
}

break;

case 5: strcpy(mode, argv[1]);
        strcpy(inputfile, argv[2]);
        strcpy(outputfile, argv[3]);
        strcpy(presentation, argv[4]);

        if((strcmp(mode, "-b") != 0) && (strcmp(mode, "-l") != 0) &&
            (strcmp(mode, "-c") != 0)) {

            printf("Error: argumento de conversion de formato no valido.\n");
            printf("Utilice: hsr2cgm [-b|-l|-c] fuente[.hsr] [destino] [presentacion]\n");
            printf("Programa terminado.\n");

            exit(1);
        }

        if (strchr(inputfile, '.') != NULL) {
            if (strstr(inputfile, ".HSR") == NULL && strstr(inputfile, ".HSr") == NULL &&
                strstr(inputfile, ".HsR") == NULL && strstr(inputfile, ".hSR") == NULL &&
                strstr(inputfile, ".Hsr") == NULL && strstr(inputfile, ".hSr") == NULL &&
                strstr(inputfile, ".hsR") == NULL && strstr(inputfile, ".hsr") == NULL) {

                printf("Error en la especificacion del archivo de entrada.\n");
                printf("Utilice: hsr2cgm [-b|-l|-c] fuente[.hsr] [destino] ");
                printf("[presentacion]\n");
            }
        }

```

```

        printf("Programa terminado.\n");
        exit(1);
    }
} else
    strcat(inputfile, ".hsr");

if (strchr(outputfile, ".") == NULL) {
    if (!strcmp(mode, "-b"))
        strcat(outputfile, ".cgm");
    else if (!strcmp(mode, "-r"))
        strcat(outputfile, ".cjk");
    else if (!strcmp(mode, "-c"))
        strcat(outputfile, ".chr");
}

if ((strcmp(presentation, "-vapfill") != 0) &&
    (strcmp(presentation, "-vanfill") != 0) &&
    (strcmp(presentation, "-wiggle") != 0) &&
    (strcmp(presentation, "-vawpfill") != 0) &&
    (strcmp(presentation, "-vawfill"))) {

    printf("Error en el parametro de presentacion de la informacion sismica \n");
    printf("Utilice. hsr2cgm [-b|-r|-c] fuente| hsr| [destino] [presentacion]\n");
    printf("Programa terminado.\n");

    exit(1);
}

break;

default:
    break;
}

/* Apertura del archivo fuente */
if ((fp1 = fopen(inputfile, "rb")) == NULL) {
    printf("No se puede abrir el archivo %s\n", inputfile);
    printf("Programa terminado.\n");

    exit(1);
}

/* Apertura del archivo destino */
if ((fp2 = fopen(outputfile, "wb")) == NULL) {
    printf("No se puede crear el archivo %s\n", outputfile);
    printf("Programa terminado.\n");

    exit(0);
}

while (!feof(fp1)) {
    #if PLATFORM
        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        command = (byte * 256) + byte2;

    #else
        fread(&command, sizeof(command), 1, fp1);

    #endif

    linethick = command >> 8;
    linethick = linethick << 12;
    linethick = linethick >> 12;
}

```

```

command = command << 8;
command = command >> 8;

/* La siguiente estructura procesa el comando del HSR-11A leído, de acuerdo al
/* rango de valores validos, o bien detecta inconsistencias en los datos de entrada */

switch(command) {
  case 0: NoOperation();
    break;
  case 1: SeismicTraces(linethick, mode, presentation, fp1, fp2);
    break;
  case 2: TimingLine(fp1, fp2);
    break;
  case 3: SerialData(fp1, fp2);
    break;
  case 4: SeismicFlush(fp1, fp2);
    break;
  case 5: RestoreRasterBuffer(fp1, fp2);
    break;
  case 6: GraphicFlush(mode, fp1, fp2);
    break;
  case 7: SectoredVectors(linethick, mode, fp1, fp2);
    break;
  case 8: ScanLineAdvance(fp1, fp2);
    break;
  case 9: Initialize(mode, fp1, fp2);
    break;
  case 10: Alphanumerics(fp1, fp2);
    break;
  case 11: Resampling(fp1, fp2);
    break;
  default: printf("Error en el archivo de entrada.\n"),
    printf("Inconsistencia de comandos del HSR-11A \n");
    printf("Programa terminado \n"),

    fclose(fp2);
    remove(outputfile);
    fclose(fp1),

    exit(1);
    break,
}

fclose(fp2), /* Se cierra el archivo de salida */
fclose(fp1), /* Se cierra el archivo de entrada */
}

```

```

/* SECTORDV.C, modulo secundario que contiene la funcion SectoredVectors() */
/*
/* Proposito: este modulo escribe directamente al archivo de salida para codificar
/* en formato cgm informacion grafica vectorizada
/*
/* Ejemplo de llamada: SectoredVectors(linethick, mode, fp1, fp2)
/*
/* Entradas: ancho de linea de la traza sismica, forma de codificacon del archivo
/* de salida cgm y apuntadores a los archivos de entrada y salida de datos.
/*
/* Salidas: dependiendo de la forma de codificacion cgm que haya sido especificada,
/* este modulo produce como salida una codificacion en forma binaria, en
/* modo de texto o en modo caracter de los segmentos de linea recta (vectores)
/* que componen la grafica.
/*
/* Subrutinas referenciadas: WriteHeader() y EncodeCharacterNumber()
/*
/* Datos pertinentes: modulo desarrollado en lenguaje de programacion C,
/* apegado al estandar ANSI-C 1969 para ser ejecutado en
/* ambiente UNIX así como de MS-DOS. Este programa forma
/* parte de un proyecto de investigacion para la Gerencia
/* de Prospeccion Geofisica del Instituto Mexicano del Petroleo (IMP).
/*
/* Autor: Miguel Angel Avila Martinez
/*
/* Fecha: 5/Junio/1998
/*
/* Modificaciones:
/*
/*

#include <stdio.h>
#include <string.h>

#include "hsr2cgm.h"

/* Variables globales que se comparten con la funcion GraphicFlush() */
short int xmin, ymin, xmax, ymax;

void SectoredVectors(unsigned short int linethick, char *mode, FILE *fp1, FILE *fp2) {
    unsigned short int word, word2;
    short int wcount, xstart, ystart, xend, yend,
              xtmp, ytmp, firstline, coords, i, j;
    unsigned char byte, byte2;
    static char flag = 1;

    if(flag) {
        xmin = ymin = xmax = ymax = 0;
        flag = 0;
    }

    /* Codificacion binaria */
    if(strcmp(mode, "-b") == 0) {
        #if PLATFORM
            fread(&byte, sizeof(byte), 1, fp1);
            fread(&byte2, sizeof(byte2), 1, fp1);

            wcount = (byte * 256) + byte2,

        #else
            fread(&wcount, sizeof(wcount), 1, fp1);

        #endif

        /* Bytes no utilizados */
        #if PLATFORM
            fread(&byte, sizeof(byte), 1, fp1);
            fread(&byte2, sizeof(byte2), 1, fp1);

```

```

i = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

j = (byte * 256) + byte2;

#else
fread(&i, sizeof(i), 1, fp1);
fread(&j, sizeof(j), 1, fp1);

#endif

/* Escritura del elemento linewidth */
WriteHeader(5, 3, 4, fp2);
byte = 0x00;
fwrite(&byte, sizeof(byte), 1, fp2);
byte = (unsigned char) linewidth;
fwrite(&byte, sizeof(byte), 1, fp2);
byte = 0x00;
fwrite(&byte, sizeof(byte), 1, fp2);
byte = 0x00;
fwrite(&byte, sizeof(byte), 1, fp2);

/* Escritura del elemento disjoint polyline */
WriteHeader(4, 2, 31, fp2);

wcount = wcount - 2;
i = (wcount - 1) * 4;

#if PLATFORM
word = i << 8;
word = word >> 8;
word2 = i >> 8;

i = word2 + word * 256;

#endif

fwrite(&i, sizeof(i), 1, fp2);

wcount = wcount - 1;
wcount = wcount / 4;

for(i = 1; i <= wcount; ++i) {
    #if PLATFORM
        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        xstart = (byte * 256) + byte2;

        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        ystart = (byte * 256) + byte2;

        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        xend = (byte * 256) + byte2;

        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        yend = (byte * 256) + byte2;
    #else
        fread(&xstart, sizeof(xstart), 1, fp1);
        fread(&ystart, sizeof(ystart), 1, fp1);
    #endif
}

```



```

        fread(&xend, sizeof(xend), 1, fp1);
        fread(&yend, sizeof(yend), 1, fp1);

#ifdefif

if(xstart > xmax)
    xmax = xstart;
else if(xstart < xmin)
    xmin = xstart;

if(ystart > ymax)
    ymax = ystart;
else if(ystart < ymin)
    ymin = ystart;

if(xend > xmax)
    xmax = xend;
else if(xend < xmin)
    xmin = xend;

if(yend > ymax)
    ymax = yend;
else if(yend < ymin)
    ymin = yend;

#ifdef PLATFORM
word = xstart << 8;
word = word >> 8;
word2 = xstart >> 8;
xstart = word2 + word * 256;

word = ystart << 8;
word = word >> 8;
word2 = ystart >> 8;
ystart = word2 + word * 256;

word = xend << 8;
word = word >> 8;
word2 = xend >> 8;
xend = word2 + word * 256;

word = yend << 8;
word = word >> 8;
word2 = yend >> 8;
yend = word2 + word * 256;

#endifif

fwrite(&xstart, sizeof(xstart), 1, fp2);
j = 0x0000;
fwrite(&j, sizeof(j), 1, fp2);

fwrite(&ystart, sizeof(ystart), 1, fp2);
j = 0x0000;
fwrite(&j, sizeof(j), 1, fp2);

fwrite(&xend, sizeof(xend), 1, fp2);
j = 0x0000;
fwrite(&j, sizeof(j), 1, fp2);

fwrite(&yend, sizeof(yend), 1, fp2);
j = 0x0000;
fwrite(&j, sizeof(j), 1, fp2);
}

/* Codificación en modo de texto */

} else if(strcmp(mode, "-t") == 0) {
#ifdef PLATFORM

```

```

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

wcount = (byte * 256) + byte2;

#else
fread(&wcount, sizeof(wcount), 1, fp1);

#endif

/* Bytes no utilizados */
#if PLATFORM
fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

i = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

j = (byte * 256) + byte2;

#else
fread(&i, sizeof(i), 1, fp1);
fread(&j, sizeof(j), 1, fp1);

#endif

wcount = wcount - 3;
wcount = wcount / 4;

firstline = coords = 0;

fprintf(fp2, " %% informacion vectorial de la grafica %%\n");
fprintf(fp2, "%v\n");

for(i = 1; i <= wcount; ++i) {
    #if PLATFORM
fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

xstart = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

ystart = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

xend = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

yend = (byte * 256) + byte2;

    #else
fread(&xstart, sizeof(xstart), 1, fp1);
fread(&ystart, sizeof(ystart), 1, fp1);
fread(&xend, sizeof(xend), 1, fp1);
fread(&yend, sizeof(yend), 1, fp1);

    #endif

    if(xstart > xmax)
        xmax = xstart;
    else if(xstart < xmin)
        xmin = xstart;
}

```

```

if(ystart > ymax)
    ymax = ystart;
else if(ystart < ymin)
    ymin = ystart;

if(xend > xmax)
    xmax = xend;
else if(xend < xmin)
    xmin = xend;

if(yend > ymax)
    ymax = yend;
else if(yend < ymin)
    ymin = yend;

if(i == 1) {
    fprintf(fp2, " linewidth %d;\n", linewidth);
    fprintf(fp2, "\n");
    fprintf(fp2, " DLSJTLINE ");
}

if(firstline == 0) {
    fprintf(fp2, "(%d,%d) (%d,%d)", xstart, ystart, xend, yend);
    firstline = 1;
    coords = 1;

    if(i == wcount)
        fprintf(fp2, "\n");
} else
    switch(coords) {
        case 0: fprintf(fp2, "(%d,%d) (%d,%d)",
                       xstart, ystart, xend, yend);
                coords = 1;

                if(i == wcount)
                    fprintf(fp2, "\n");

                break;
        case 1: fprintf(fp2, "(%d,%d) (%d,%d)",
                       xstart, ystart, xend, yend);
                coords = 2;

                if(i == wcount)
                    fprintf(fp2, "\n");

                break;
        case 2: fprintf(fp2, "(%d,%d) (%d,%d)",
                       xstart, ystart, xend, yend);
                coords = 0;

                if(i == wcount)
                    fprintf(fp2, "\n");
                else
                    fprintf(fp2, "\n");

                break;
        default: break;
    }
}

fprintf(fp2, "\n");
fprintf(fp2, "\n");

/* Codificación en modo caracter */
} else if(strcmp(mode, "-c") == 0) {
    #if PLATFORM

```

```

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

wcount = (byte * 256) + byte2;

#else
fread(&wcount, sizeof(wcount), 1, fp1);

#endif

/* Bytes no utilizados */
#if PLATFORM
fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

i = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);

j = (byte * 256) + byte2;

#else
fread(&i, sizeof(i), 1, fp1);
fread(&j, sizeof(j), 1, fp1);

#endif

wcount = wcount - 3;
wcount = wcount / 4;

/* Escritura del elemento disjoint polyline */
byte = 0x21;
fwrite(&byte, sizeof(byte), 1, fp2);

xtmp = 0;
ytmp = 0;

for(i = 1; i <= wcount, ++i) {
    #if PLATFORM
        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        xstart = (byte * 256) + byte2;

        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        ystart = (byte * 256) + byte2;

        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        xend = (byte * 256) + byte2;

        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);

        yend = (byte * 256) + byte2;

    #else
        fread(&xstart, sizeof(xstart), 1, fp1);
        fread(&ystart, sizeof(ystart), 1, fp1);
        fread(&xend, sizeof(xend), 1, fp1);
        fread(&yend, sizeof(yend), 1, fp1);

    #endif

    #endif

    if(xstart > xmax)
        xmax = xstart;

```

```
else if(xstart < xmin)
    xmin = xstart;

if(ystart > ymax)
    ymax = ystart;
else if(ystart < ymin)
    ymin = ystart;

if(xend > xmax)
    xmax = xend;
else if(xend < xmin)
    xmin = xend;

if(yend > ymax)
    ymax = yend;
else if(yend < ymin)
    ymin = yend;

if(i == 1) {
    EncodeCharacterNumber(xstart, fp2);
    EncodeCharacterNumber(ystart, fp2);
} else {
    EncodeCharacterNumber(xstart - xtmp, fp2);
    EncodeCharacterNumber(ystart - ytmp, fp2);
}

EncodeCharacterNumber(xend - xstart, fp2);
EncodeCharacterNumber(yend - ystart, fp2);

xtmp = xend;
ytmp = yend;
}
}
}
```

```

/* SEISMICT.C, modulo secundario que incluye el codigo de la funcion SeismicTraces() */
/*
/* Proposito: este modulo inicia el proceso de los comandos 'Sesimic Traces' a partir del
/* flujo de datos de entrada. La funcion lee y procesa algunos parametros
/* que definen atributos de las trazas sismicas y posteriormente canaliza el
/* flujo del programa hacia una de las cinco funciones que se encargan de
/* codificar el archivo cgm, dependiendo del modo de despliegue especificado.
/*
/* Ejemplo de llamada: SeismicTraces(finethick, mode, presentation, fp1, fp2)
/*
/* Entradas: ancho de linea de las trazas sismicas, forma de codificacion del
/* archivo de salida cgm, modo de despliegue de las trazas y apuntadores
/* a los archivos de entrada y salida de datos.
/*
/* Salidas: ninguna, este modulo no escribe en el archivo de salida, solo constituye
/* un paso intermedio en el procesamiento de la Informacion sismica
/*
/* Subrutinas referenciadas: ScalingF16Bits(), VarAreaPosFill(),
/* VarAreaNegfill(), WiggleOnly(),
/* VarAreaWigglePosFill(), VarAreaWiggleNegFill()
/*
/* Datos pertinentes: modulo desarrollado en lenguaje de programacion C,
/* apegado al estandar ANSI-C 1989 para ser ejecutado en
/* ambiente UNIX asi como de MS-DOS. Este programa forma
/* parte de un proyecto de investigacion para la Gerencia
/* de Prospeccion Geofisica del Instituto Mexicano del Petroleo (IMP).
/*
/* Autor: Miguel Angel Avila Martinez
/*
/* Fecha: 5/Junio/1998
/*
/* Modificaciones:
/*
/*

```

```

#include <math.h>
#include <stdio.h>
#include <string.h>

#include "hsr2cgm.h"

```

```

/* Variables globales comunes a las cinco funciones que procesan informacion sismica */

psnode *head;

float trascal;
unsigned short int wcount, displaymode, inputformat, interpolation,
trascalarf, trascalr2, posmax, negmax, xstart,
ystart;

signed char fill;

void SeismicTraces(short int linethick, char *mode, char *presentacion, FILE *fp1, FILE *fp2) {

    unsigned short int i;
    unsigned char byte, byte2;

    /* Las siguientes instrucciones se encargan de leer los parametros wcount,
    /* displaymode, inputformat, interpolation, trascal, posmax, negmax,
    /* xstart y ystart del archivo de entrada. Los valores de trascal son
    /* procesados a su vez por medio de la funcion ScalingF16Bits(), mientras
    /* que los restantes valores se utilizan posteriormente en la codificacion de las trazas. */

    #if PLATFORM
        fread(&byte, sizeof(byte), 1, fp1);
        fread(&byte2, sizeof(byte2), 1, fp1);
        wcount = (byte * 256) + byte2;
    #endif

```

```

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
displaymode = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
inputformat = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
interpolation = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
i = (byte * 256) + byte2;

#else
fread(&wcount, sizeof(wcount), 1, fp1);
fread(&displaymode, sizeof(displaymode), 1, fp1);
fread(&inputformat, sizeof(inputformat), 1, fp1);
fread(&interpolation, sizeof(interpolation), 1, fp1);
fread(&i, sizeof(i), 1, fp1);

#endif

switch(inputformat) {
case 0:
case 1:
    #if PLATFORM
fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
tracescalar1 = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
tracescalar2 = (byte * 256) + byte2;

tracescalar = ScalingF16Bits(tracescalar1, tracescalar2);

#else
fread(&tracescalar1, sizeof(tracescalar1), 1, fp1);
fread(&tracescalar2, sizeof(tracescalar2), 1, fp1);
tracescalar = ScalingF16Bits(tracescalar1, tracescalar2);

#endif
break;
case 2:
break;
case 3:
break;
default:
break;
}

#if PLATFORM
fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
i = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
posmax = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);
fread(&byte2, sizeof(byte2), 1, fp1);
negmax = (byte * 256) + byte2;

fread(&byte, sizeof(byte), 1, fp1);

```

```

    fread(&byte2, sizeof(byte2), 1, fp1);
    xstart = (byte * 256) + byte2;

    fread(&byte, sizeof(byte), 1, fp1);
    fread(&byte2, sizeof(byte2), 1, fp1);
    ystart = (byte * 256) + byte2;

#else
    fread(&i, sizeof(i), 1, fp1);
    fread(&posmax, sizeof(posmax), 1, fp1);
    fread(&negmax, sizeof(negmax), 1, fp1);
    fread(&xstart, sizeof(xstart), 1, fp1);
    fread(&ystart, sizeof(ystart), 1, fp1);

#endif

fread(&byte, sizeof(byte), 1, fp1);
fread(&fill, sizeof(fill), 1, fp1);

displaymode = displaymode << 8;
displaymode = displaymode >> 8;
wcount = wcount - 12;

/* Si el parametro 'presentacion' fue especificado en la linea de ordenes del
/* programa, la variable displaymode se ajusta al modo de despliegue deseado.
/* en caso contrario, se utiliza la forma de despliegue por defecto que viene
/* indicada en el archivo original de entrada */

if(presentacion[0] == '0')
;

else {
    if(!strcmp(presentacion, "-vapfill"))
        displaymode = 0;
    else if(!strcmp(presentacion, "-vanfill"))
        displaymode = 1;
    else if(!strcmp(presentacion, "-wiggles"))
        displaymode = 2;
    else if(!strcmp(presentacion, "-vawpfill"))
        displaymode = 3;
    else if(!strcmp(presentacion, "-vawnfill"))
        displaymode = 4;
}

/* Por último, se invoca a una de las funciones que codifican directamente
/* información sísmica dependiendo del valor que haya sido asignado a 'displaymode' */

switch(displaymode) {
    case 0: VarAreaPosFill(linethick, mode, fp1, fp2);
        break;
    case 1: VarAreaNegFill(linethick, mode, fp1, fp2);
        break;
    case 2: WiggleOnly(linethick, mode, fp1, fp2);
        break;
    case 3: VarAreaWigglePosFill(linethick, mode, fp1, fp2);
        break;
    case 4: VarAreaWiggleNegFill(linethick, mode, fp1, fp2);
        break;
    default
        break;
}
}
}

```



### 7.3 Implementación y prueba final del sistema.

Una vez concluida la etapa de codificación, el paso final consiste en instalar el programa en la máquina destino donde será accesible a los usuarios para su ejecución. Dicha máquina es en este caso el nodo 21 del sistema paralelo IBM SP2 9076, dentro de la ruta del subdirectorio */usr/local/bin*. Manteniendo una copia de los programas fuente en un subdirectorio temporal dentro de este sistema, la compilación de los módulos se lleva a cabo de la siguiente manera a partir de la línea de comandos del sistema operativo AIX:

```
$xlc *.c -lm -v -o/usr/local/bin/hsr2cgm
```

```
alphanum.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,alphanum.c,alphanum.o,alphanum.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
encchnum.c.
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,encchnum.c,encchnum.o,encchnum.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
graphicf.c
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,graphicf.c,graphicf.o,graphicf.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
hsr2cgm.c.
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,hsr2cgm.c,hsr2cgm.o,hsr2cgm.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
if16bits.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,if16bits.c,if16bits.o,if16bits.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
initial.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,initial.c,initial.o,initial.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
nooper.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,nooper.c,nooper.o,nooper.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
pull.c.
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,pull.c,pull.o,pull.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
push.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,push.c,push.o,push.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
resampln.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,resampln.c,resampln.o,resampln.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
restorf.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,restorf.c,restorf.o,restorf.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
scalingf.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,scalingf.c,scalingf.o,scalingf.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
scanlad.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,scanlad.c,scanlad.o,scanlad.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
sectordv.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,sectordv.c,sectordv.o,sectordv.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
seismicf.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,seismicf.c,seismicf.o,seismicf.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
seismict.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,seismict.c,seismict.o,seismict.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
serial.c:
exec: /usr/lpp/xlc/bin/xlcentry(xlcentry,serial.c,serial.o,serial.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qno,-qnoconst,NULL)
```

```

timmigl.c:
exec: /usr/app/xic/bin/xcentry(xcentry,timmigl.c,timmigl.o,timmigl.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)
vanfill.c:
exec: /usr/app/xic/bin/xcentry(xcentry,vanfill.c,vanfill.o,vanfill.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)
vapfill.c:
exec: /usr/app/xic/bin/xcentry(xcentry,vapfill.c,vapfill.o,vapfill.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)
vawmfill.c:
exec: /usr/app/xic/bin/xcentry(xcentry,vawmfill.c,vawmfill.o,vawmfill.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)
vawpfill.c:
exec: /usr/app/xic/bin/xcentry(xcentry,vawpfill.c,vawpfill.o,vawpfill.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)
wheader.c:
exec: /usr/app/xic/bin/xcentry(xcentry,wheader.c,wheader.o,wheader.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)
wiggleon.c:
exec: /usr/app/xic/bin/xcentry(xcentry,wiggleon.c,wiggleon.o,wiggleon.lst,-D_IBMR2,-D_AIX,-D_AIX32,-qlanglvl=extended,-qnor,-
qnorconst,NULL)

exec: /bin/Ad(jd,-H512,-T512,-bhalt:4-
o,hsr2cgm,/lib/crt0.o,alphanum.o,encchnum.o,graphicf.o,hsr2cgm.o,if16bits.o,initial.o,nooper.o,pull.o,push.o,resampln.o,restorf.o,sca
lingf.o,scanlad.o,sectordv.o,seismicf.o,seismicf.o,seriald.o,timmigl.o,vanfill.o,vapfill.o,vawmfill.o,vawpfill.o,wheader.o,wiggleon.o,-lm,-
lc,NULL)

unlink: alphanum.o
unlink: encchnum.o
unlink: graphicf.o
unlink: hsr2cgm.o
unlink: if16bits.o
unlink: initial.o
unlink: nooper.o
unlink: pull.o
unlink: push.o
unlink: resampln.o
unlink: restorf.o
unlink: scalingf.o
unlink: scanlad.o
unlink: sectordv.o
unlink: seismicf.o
unlink: seismicf.o
unlink: seriald.o
unlink: timmigl.o
unlink: vanfill.o
unlink: vapfill.o
unlink: vawmfill.o
unlink: vawpfill.o
unlink: wheader.o
unlink: wiggleon.o

```

La ejecución del sistema se lleva a cabo según lo expuesto en el Manual de Usuario Preliminar expuesto en el inciso 6.3.3. A continuación se describen los resultados obtenidos al ser probado con datos reales de entrada

### 7.3.1 Ejemplos de las gráficas de secciones sísmicas generadas por el sistema de conversión.

En las siguientes páginas se ilustra, a escala reducida, el aspecto final de las secciones sísmicas una vez que han sido creadas a partir del programa de conversión. El tamaño real de cada sección es de aproximadamente 80 centímetros de ancho y una longitud variable, dependiendo de la cantidad de información sísmica que haya sido procesada, que oscila entre 1.5 y tres metros de largo.

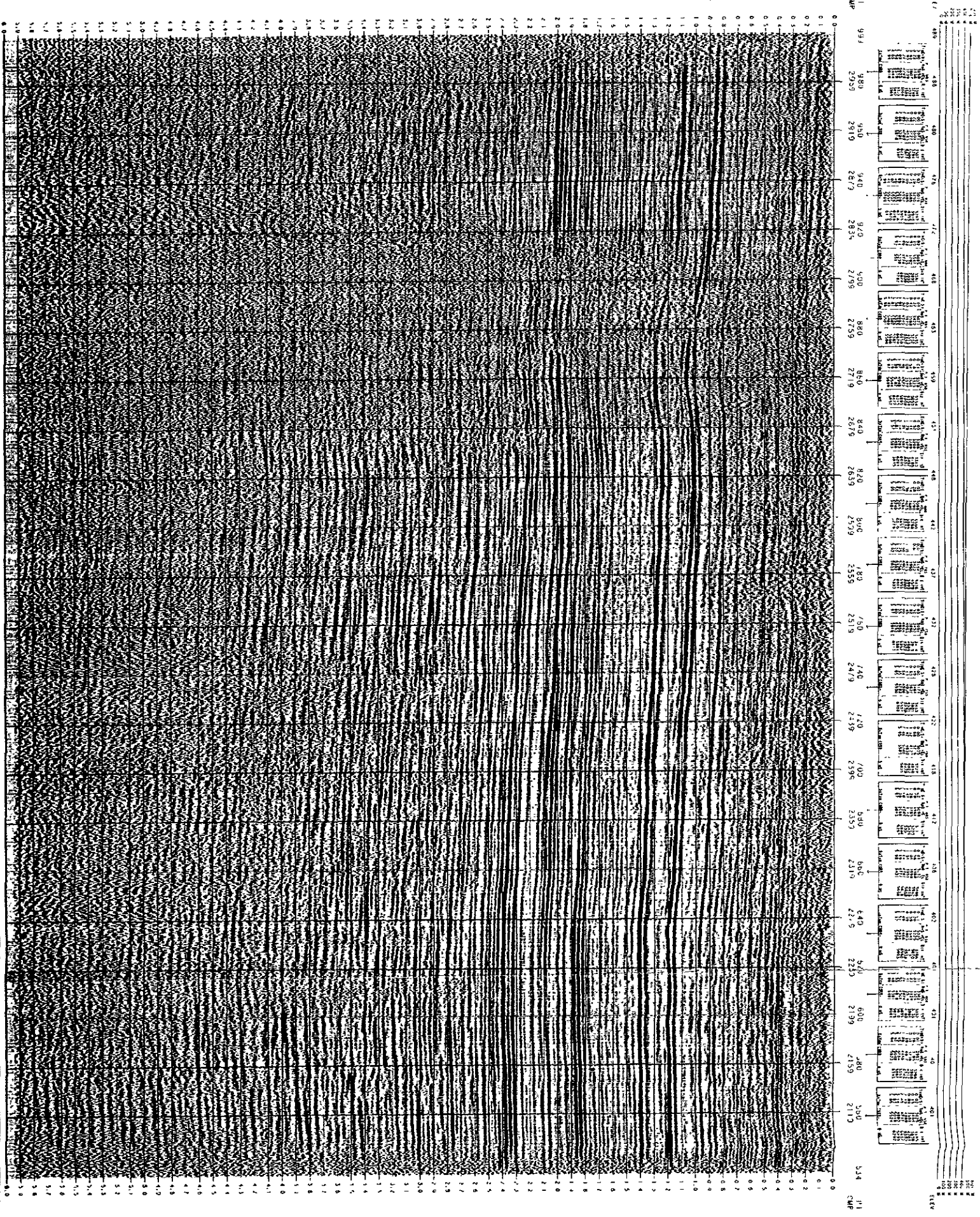
Toda sección sísmica generada dentro del IMP se distingue por tener una estructura uniforme. Vista la gráfica de frente, se pueden observar en la parte superior un conjunto de líneas paralelas horizontales donde se dibuja un esquema de las elevaciones correspondientes a la zona en donde se hizo el estudio de prospección, debajo de esta se encuentran las tablas de velocidades para varias trazas sísmicas en puntos CDP específicos. En la parte central de la gráfica, y abarcando la mayor parte de la misma, se puede observar la información sísmica representada en una de las tres modalidades de despliegue según lo visto en el capítulo 2. En el borde inferior de la gráfica se muestra un histograma del apilamiento correspondiente.

Finalmente, en el extremo derecho de la sección sísmica se puede observar lo que comúnmente se conoce como *sello* de la gráfica. Esta es un área que tiene como propósito informar al intérprete dónde se ubica el prospecto de estudio (en la parte superior: línea, área, zona, grupo, etc.); los datos de campo (tipo de energía utilizada, número de pozos realizados, tiempo de grabación, largo del tendido, fecha en que fueron registrados los datos); la secuencia de proceso y parámetros (es decir, el proceso hecho a la información sísmica con el auxilio de los programas y sistemas de cómputo); los símbolos utilizados, las escalas horizontales, verticales, fecha de proceso y el nombre del analista responsable.

Las siguientes tres gráficas corresponden a secciones sísmicas pertenecientes a la línea 51 del área Lampazos Monclova, prospecto Minero, ubicado en la zona noreste de la República Mexicana (ver mapa de la División Regional de Pemex en el capítulo 3, página 24). En la primera de ellas se muestra la información sísmica en la modalidad de presentación que incluye el wiggle, el área variable y un relleno positivo de la traza. La segunda muestra únicamente las trazas del registro sísmico (wiggle) y en la tercera exclusivamente el área variable en su parte positiva.

L-51 NW

SE L-51



LINER 51 DE P.T. 999 A P.T. 534  
 AREA LAMPAZOS - MONCLOVA  
 PROSPECTO MINERO NESY-25N R. 300.0 M  
 ZONA NOROESTE GRUPO

PEMEX  
 CENTRO DE INVESTIGACIÓN CIENTÍFICA  
 SUBTERRÁNEA DE EXPLORACIÓN DE TECNOLOGÍA  
 DE EXPLORACIÓN

**DATOS DE CAMPO**

EQUIPO: MDS-13 ENERGIA: Y BRONSON  
 FILTROS: 9-86 12 NO. DE FOLIOS:  
 MANIFIESTA: I.F.P. PROF. MED.A:  
 PARA ES DE GAS: 95 FASMA MED.A:  
 TIPO DE GRAB.: 6 S D.S.I./T.: 200 W  
 GRABACION: M/S DETEC. POR GRUPO: 12  
 V.B.: 3000 M/S D.S.I./GRUPOS: 38  
 TIPO DE BIL: 2650-3000-0W DETEC. TIPO: LES-1000  
 COMP. INVA. GEOL. VALUACIONES: FECHA: JUNIO 1987

**SECUENCIA DE PROCESO Y PARAMETROS**

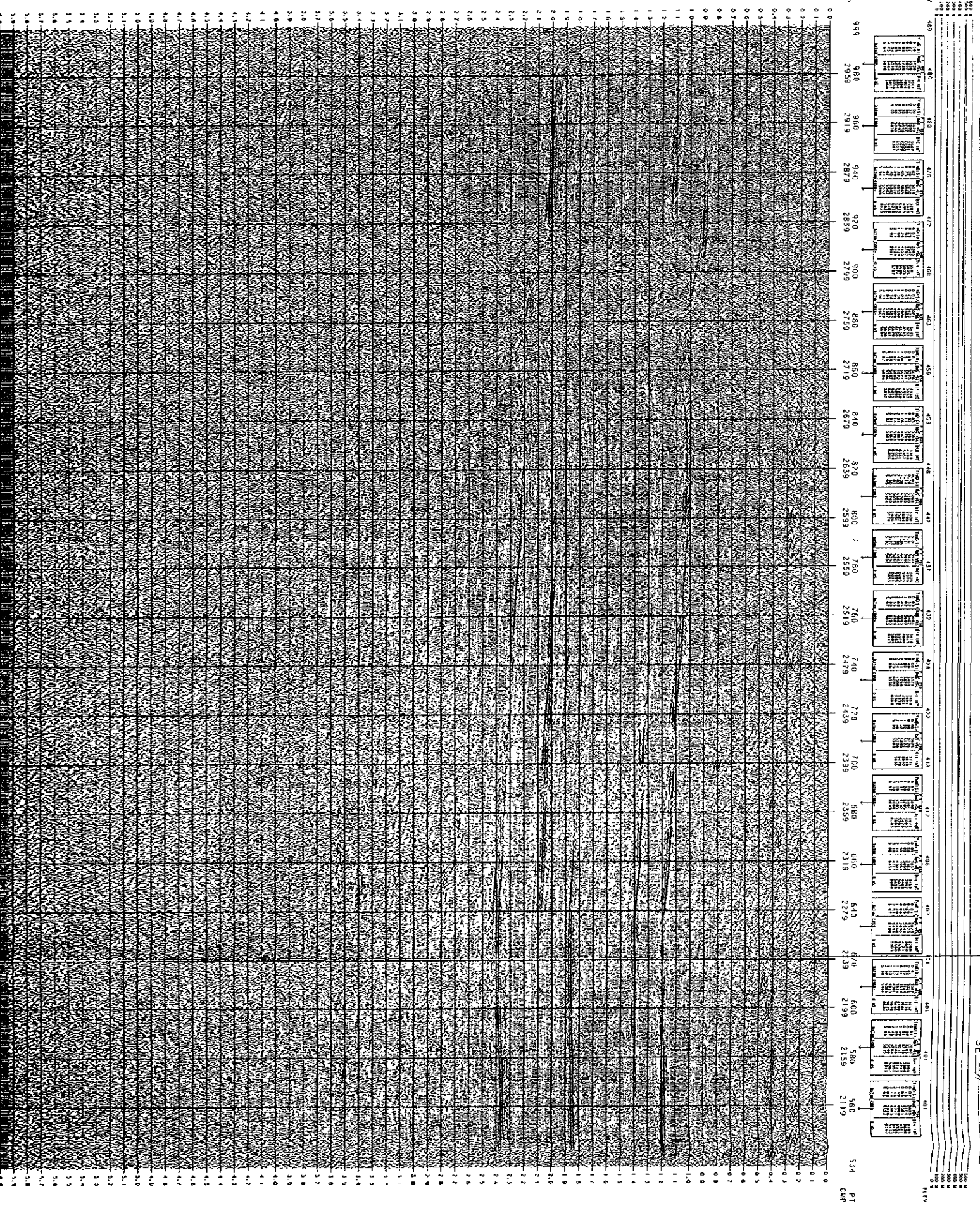
EUROPE 380450 SP 2 (FRANCA)  
 DE UTILIZADO:  
 ARREGLO GEOMERLO  
 LIBRACION DE ERVAS  
 NOMBRADO DE ARCHIVOS AMERICAS  
 COMPRESION POR DIFERENCIA ESPECIFICA  
 DECHOVA CON COM. S. EN A. S. O. R. C. I. E  
 MEDIANA D. STANCA PRECISIVA: 34 MS  
 CONJUNTO DEL ORDENADOR  
 INHIBICION DE VELOCIDADES (S. S. H. M. A. N. A. )  
 VERIFICACION ALTERNATIVA DE VELOCIDADES  
 CORRECCION DE ESTADISTICAS RESIDUALES  
 APLICADO 12007  
 LAS CANTIDAD DE PROBLEMA: 100 5121 210 110 400;  
 COMPRESION DE ARCHIVOS (RASC)

FILTRADO: 0-0  
 \* LARGO (SEC) FILTROS (HZ)  
 0.00 - 0.80 12 - 40  
 0.80 - 4.00 10 - 35  
 4.00 - 6.00 8 - 30

\*\*\* SECCION NORMAL CON 117 P.I. PROCESADOS \*\*\*

<p><b>LEYENDA</b></p> <ul style="list-style-type: none"> <li>PROB. DE GAS</li> <li>PROB. ACIFR. GAS</li> <li>PROB. VIL.</li> <li>PROB. VIL.</li> </ul>	<p><b>FECHA DE PROCESO:</b>          JULIO 1988</p> <p><b>PROCESADO POR:</b>          CARLOS GONZALEZ</p>
--	---





LINER 51 DE P.T. 999 A P.T. 534.  
 RPER LAMPAZOS - MONCLOVA  
 PROSPECTO MINERO NESV-25M.R. 300.0 M  
 ZONR NORESTE GRUPO



**DATOS DE CAMPO**

EQUIPO: MOS-10 ENERGIA: VIBROSIENDO  
 FILTROS: 9-86 HZ NO. DE PÓZOS:  
 GANANCIA: L.F.P. PROF. MEDIA: M  
 CANALES DE GRAB.: 96 CARGA MEDIA: KG  
 TIEMPO DE GRAB.: 6 S DIST./P.T.: 200 M  
 GRABACION: DIST./DETECTORES: 50 M  
 VO.: W/S DETEC. POR GRUPO: 12  
 VII: 3000 W/S DIST./GRUPOS: 5M  
 INDIIC-BII 2550 3000 OH DIFEC. 1190: IRS 1000  
 COMPANIA: GEDEVALUACIONES FECHA: JUNIO 1989

**SECUENCIA DE PROCESO Y PARAMETROS**

FOUPO DE PROCESO SP-7 (PROMAX)  
 DILUI LIXADO.  
 ARRISO GOU LITIO  
 COLOCION DE TRAZAS  
 BORRADO DE PRUEBAS ANEBROS  
 COMPENSACION POR DIVERGENCIA ESFERICA  
 DECONVOLUCION CONSISTENTE CON LA SUPERFICIE  
 MINIMA DISTANCIA PREDICTIVA : 34 MS  
 LONGITUD DEL OPERADOR : 200 MS  
 INTERPRETACION DE VELOCIDADES (SEMANA)  
 INTERPRETACION INTERACTIVA DE VELOCIDADES (V1)  
 CORRECCION DE ESTADISTICAS RESIDUALES  
 APLIADO 1700Z  
 \*\*\* FOUPO DE PROCESO : IBM 9121 210 (10 MEG. I  
 COMPENSACION DE AMPLITUDOS (INAC)

**ILLINO PROMEDIO**

ILLINO (SIC)	ILLINOS (L2)
0.00	12 - 40
0.80	10 35
4.00	8 30

\*\*\* SECCION NORMAL CON 117 P.T. PROCESADOS \*\*\*

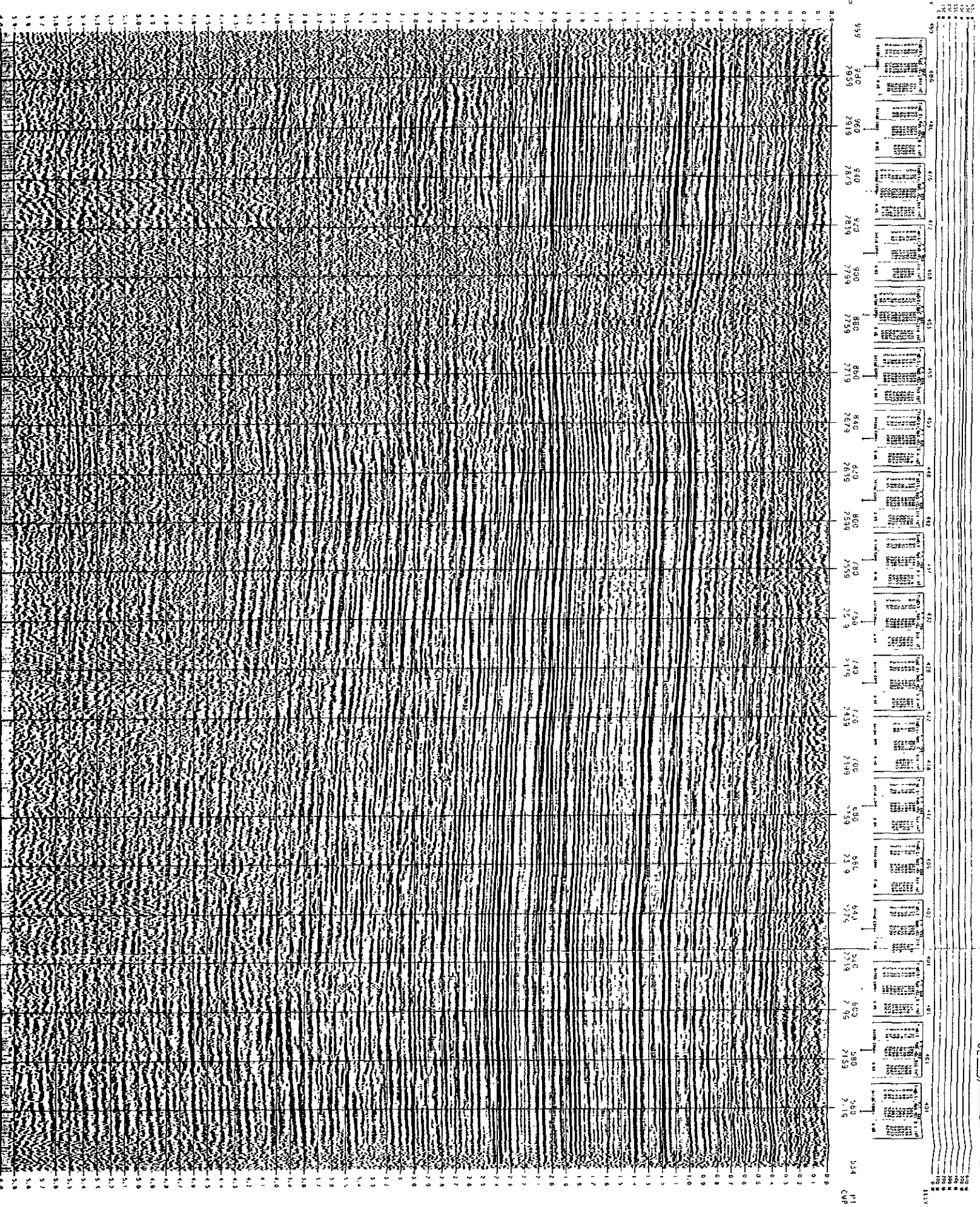
**SIMBOLOS**

- FIRM. M. CAS
- FIRM. ALBERT CAS
- OPORTO
- ANCL. VIL.

**ESCALA**  
 HORIZONTAL 1:10000  
 VERTICAL 1:10000

**FECHA DE PROCESO:**  
 JULIO 1989

**PROCESADO POR:**  
 CARLOS DOMALIZ



LINEA 51 DE P.T. 999 A P.T. 334  
 AREA LAMPAZOS - MONCLOVA  
 PROYECTO MINERO NESV-25N R. 300.0 M  
 ZONA NORESTE GRUPO NESV-25N R. 300.0 M



DAIOS DE CAMPO

CORTO:	405-10	ENERGIA:	VIBROSISMO
FILTROS:	9-85 HZ	NO. DE PULSOS:	M
CANALIZACION:	I.F.F.	PROG. METRO:	M
CANALES DE SP-8:	96	CARGA MEDIA:	KG
TIPO DE CABLE:	6.5	REC./P.T.:	209 M
ORIENTACION:		DIST./DETECCIONES:	50 M
		DETEC. POR GRUPO:	12
		DIST./GRUPOS:	5M
		DETEC. TIPO:	ERS-1000
		COMPANIA DE SERVICIOS:	INDIO 1484

SECUENCIA DE PROCESO Y PARAMETROS

GRUPO DE PROCESO S-7 (SP-8) (S-7)

DE QUE TIPO: ABRIL 1980

ABRIL 1980

EDICION DE TRAZAS

BORRADO DE PRIMEROS ARRASTES

COMPENSACION POR DIVERGENCIA ESTEREA

DEFORMACION CON COMPENSACION EN LA SUPERFICIE

QUEDA DISTANCIAS PERPENDICULARES

LOCACION DEL ORIENTADOR

INTERFERENCIA DE VECEDORES (SINUSOIDAL)

INTERFERENCIA DE VECEDORES (SINUSOIDAL)

COMPENSACION DE ESTADISTICA REPTORIAL

APLADO 1200V

\*\*\* EQUIPO DE PROCESO: IBM 9121 210 (10 MCG.)

COMPENSACION DE AMPLITUDES (BAAC)

FECHA DE PROCESO

TIPO DE SISMO

0.00 0.80 12 40

0.80 4.00 10 35

4.00 6.00 8 30

SECCION NORMAL CON 117 P.T. PROCESADOS \*\*\*

ESTRUCTURA

PROCESADO POR: JULIO 1980

CARLOS CONTRERAS

---

## Conclusiones

Al término del desarrollo del software de conversión, y una vez instalado en uno de los nodos de la computadora IBM SP2 9076, se observó que el cambio de formato HSR-11B a CGM se efectuaba de manera correcta y los archivos así generados eran interpretados como válidos sin importar su modalidad de codificación. Para la comprobación de las gráficas se utilizó un programa llamado CGMView presente en el entorno de OMEGA instalado también en la computadora IBM SP2 9076. Este programa, como ya se mencionó, es una aplicación utilizada por los analistas para el procesamiento de información sísmica.

Considero que una aportación valiosa del sistema fue la de satisfacer una necesidad concreta de intercambio de información entre dos equipos de cómputo esencialmente incompatibles entre sí, pero que mediante el uso de las interfaces de software apropiadas, es posible que puedan “convivir”, generando e intercambiando datos que se apeguen a una norma estándar, de tal manera que el otro equipo de cómputo también los pueda procesar. El desarrollo de tales interfaces en software resulta muy económico en comparación con las interfaces en hardware equivalentes, es decir, la utilización de equipo especialmente diseñado para lograr una conexión física entre dispositivos y que se encargue de la transferencia de información de una plataforma a otra.

Durante el tiempo que tengo de laborar en la Gerencia de Prospección Geofísica del IMP, me he dado cuenta que entre las grandes aplicaciones especialmente diseñadas para el procesamiento de información sísmica, existen características no cubiertas por dichos programas pero que son deseables con la finalidad de aprovechar al máximo los resultados obtenidos en un equipo de cómputo para procesarlos en otro de diferentes características. Ejemplo de lo anterior lo constituye el intercambio de información sísmica de campo, la cual debe ser puesta en un formato estándar dentro de la industria (SEG-Y, o WES-4 principalmente) con el propósito de que pueda ser posteriormente leída y procesada por programas y equipos de cómputo completamente diferentes al que los generó. Las aplicaciones generalmente ofrecen estas opciones pero en otros casos el personal de sistemas debe desarrollar módulos de software “a la medida” para cada aplicación, integrarlos en el ambiente del sistema operativo en que se ejecutan estas aplicaciones y hacerlos accesibles a los usuarios en forma de herramientas de soporte. El sistema desarrollado en la presente tesis puede ser clasificado dentro de esta última categoría.

El tiempo en que una gráfica de sección sísmica era transformada a formato CGM dependía directamente de la carga de trabajo presente en el sistema SP2. Para una carga normal, una sección de 1500 trazas se generaba aproximadamente en siete minutos.

---

Una vez visualizadas las secciones sísmicas en pantalla mediante el programa CGMView, se procedió a evaluar su aspecto físico al ser impresas por los equipos graficadores. Los resultados obtenidos en el graficador XEROX Versatec 6410 a partir de los archivos CGM generados por el programa de conversión, comparados con las gráficas impresas en el mismo graficador pero generados a partir de CAPS en el computador central IBM 9121, fueron idénticos en la información vectorizada y sólo con diferencias mínimas en la amplitud de las trazas sísmicas, casi imperceptibles a simple vista pero dentro de una tolerancia aceptable según opinión de los analistas geofísicos. El tamaño de los archivos CGM generados fue aproximadamente de 8 a 15 Mb, incrementándose esta cantidad dependiendo del número de trazas sísmicas presentes en el archivo de entrada.

Por último, para las pruebas de almacenamiento en cintas magnéticas, se obtuvo que en un sólo cartucho tipo 3480 de IBM con una capacidad de almacenamiento de 250 Mb de información, se podían almacenar 20 imágenes CGM en formato tar y de un tamaño promedio de 12Mb para cada archivo.



---

## Glosario de términos

**Acumulaciones estratigráficas.** Concentración en un determinado nivel, en forma de estratos, de sustancias arrastradas hacia abajo por las aguas de infiltración. Este nivel se denomina horizonte de acumulación u horizonte fluvial.

**Alidada.** Regla topográfica , con pínulas en los extremos.

**Anticlinal.** Parte elevada o convexa de un pliegue del terreno; puede ser una montaña, una loma o simplemente un pequeño montículo; la falda del anticlinal se llama el ala; la línea que pasa por la cima es el eje.

**Apilamiento.** Reforzamiento (suma) de las señales recibidas por el geófono producidas por el reflejo de las ondas sísmicas en las diferentes capas del subsuelo.

**Basamento.** Conjunto pétreo que dentro de sus intensas deformaciones, metamorfismos e intrusiones, presenta una gran unidad; su antigüedad es siempre anterior a la de las formaciones que soporta, llamadas coberteras, que tienen una estructura distinta.

**Buzamiento.** Inclinación de un filón o una capa de terreno.

**CDP.** Punto de reflejo común (del inglés Common Deep Point). Es el lugar sobre la superficie del suelo donde se colocan los geófonos que van a registrar las ondas sísmicas.

**CGM.** Siglas en inglés de Computer Graphics Metafile (metaarchivo para gráficos por computadora) Estándar de la ANSI/ISO el cual define un mecanismo para retener y transportar información de tipo gráfico vectorial, así como caracteres de control. Todos los archivos de este tipo se encuentran en un formato conveniente para su almacenamiento y recuperación en sistemas de cómputo, además de que describen una o más imágenes gráficas en una forma totalmente independiente del dispositivo físico (hardware) de visualización que se emplee en un sistema de cómputo.

**Contador Geiger.** Instrumento para detectar y medir la radiactividad (principalmente rayos alfa, beta y gama).

**Domo salino.** Forma que termina en un punto o línea breve desde la cual descienden sus materiales de forma brusca.

**Elemento delimitador.** Conjunto especial de elementos gráficos que sirven para señalar el inicio y final de los bloques básicos que componen todo archivo CGM. El estándar define 17 elementos de este tipo, todos ellos agrupados en la clase 0.

**Elevación.** Altura con respecto a un plano de referencia en el subsuelo hasta la superficie del terreno en donde son colocados los sismodetectores.

---

**Elevación estructural.** Medida de la altitud, sobre el nivel del mar, de un punto situado dentro de una capa del subsuelo.

**Estrato.** Cubierta. Capa de la corteza terrestre.

**Formato tar.** Modo de grabación en cintas magnéticas (*tape archive*), presente en el ambiente del sistema operativo UNIX en la forma del comando *tar*. Sirve para grabar y recuperar estructuras de archivos y subdirectorios como una entidad única, además de que permite comprimirlos con el objeto de economizar espacio de almacenamiento.

**Geófono.** Aparato que permite la transformación de energía sísmica en impulsos eléctricos que puedan medirse.

**Horizonte.** Línea lejana que a la vista separa la tierra del cielo. Espacio de la superficie terrestre limitado por esa línea

**Línea de tiro.** Trayectoria en línea recta a partir del punto de explosión y en dirección del conjunto de sismodetectores (geófonos)

**Litología.** Parte de la geología que trata de las rocas, también llamada petrología o petrografía

**Macro elemento.** Sistema basado en computadora que forma parte de un sistema basado en una computadora mayor.

**Mainframe.** Sistema de cálculo de grandes dimensiones, de propósito general que incorpora una Unidad Central de Proceso, memoria, dispositivos de E/S, fuente de alimentación y un bastidor. Típicamente se utilizan estos equipos en aquellas áreas en las que se requiere una gran capacidad de ejecución, cómputo numérico intensivo o para el manejo y gestión de grandes bases de datos.

**Mena.** Metal en bruto. Mineral metalífero, principalmente de hierro, tal y como sale de la mina

**Metaarchivo.** Dentro del formato CGM, son un tipo de archivos especiales que contienen datos gráficos así como información de control. Esta información contiene una descripción independiente de cualquier dispositivo físico (*hardware*) de una o más imágenes y constituyen un mecanismo para el fácil transporte y retención de imágenes gráficas.

**Métodos de prospección geofísica.** Conjunto de técnicas de exploración del subsuelo utilizadas dentro de la ciencia geofísica con el objeto de conocer y estudiar la condición estructural del subsuelo a grandes profundidades.

**Pínula.** Tablilla metálica, provista de una abertura longitudinal, para dirigir visuales en ciertos instrumentos topográficos y astronómicos.

**Prospección.** Exploración del subsuelo, basada en el estudio de los caracteres del terreno, para descubrir sus riquezas minerales, etc

---

**Raster.** Término que hace referencia a una línea horizontal de puntos que componen una imagen. Cada punto es conocido con el nombre de pixel, el cual puede adoptar un color único dentro de un rango (paleta) de colores fijo y preestablecido, dependiendo de las capacidades gráficas del dispositivo de visualización.

**Reluctancia.** Resistencia de un circuito al flujo magnético.

**Roca.** Conjunto de sustancias minerales que forman masas independientes y constituyen gran parte de la corteza terrestre

**Rocas ígneas.** La formada por solidificación de materia en fusión procedente del magma, por ejemplo el mármol, granito, neis, cuarzita, etc.

**Rocas metamórficas.** La procedente de una transformación causada por agentes exteriores, por ejemplo, el mármol cristalino de piedra caliza, la cuarcita de arenisca, etc.

**Rocas sedimentarias.** La formada por materiales depositados por el agua, el viento, etc. Por ejemplo la sal, arenisca, piedra caliza, arcilla, etc

**Sistema.** 1. Conjunto u ordenación de cosas relacionadas de tal manera que forman una unidad o un todo orgánico; 2 Un conjunto de hechos, principios, reglas, etc., clasificados y ordenados de tal manera que muestran un plan lógico uniendo las diferentes partes, 3 Un método o plan de clasificación u ordenación; 4. Una forma establecida de hacer algo: un método, un procedimiento, etc.

**Trampas estructurales.** Formación rocosa debajo de una capa impermeable donde el petróleo se acumula. Normalmente compuestas de rocas porosas (arcilla, serie evaporítica, etc.) se presenta en variadas estructuras: gran cúpula anticlinal, bisel bajo una discordancia, algunas estructuras falladas o diapíricas, dispositivos lenticulares, etc.

**Wiggle.** Término inglés que literalmente significa mover o moverse rápidamente de un lado a otro. En el contexto que nos ocupa, se refiere a la traza sísmica tal y como es registrada a partir de los sismodetectores.

---

## Bibliografía

1. American National Standards Institute/International Standards Organization.  
ANSI X3.122/ISO 8632. Information Processing Systems - Computer Graphics.  
Metafile for the storage and transfer of picture information.  
Part 1: Functional description. New York
2. American National Standards Institute/International Standards Organization  
ANSI X3.122/ISO 8632. Information Processing Systems - Computer Graphics.  
Metafile for the storage and transfer of picture information.  
Part 2: Character encoding. New York
3. American National Standards Institute/International Standards Organization  
ANSI X3.122/ISO 8632 Information Processing Systems - Computer Graphics.  
Metafile for the storage and transfer of picture information  
Part 3: Binary encoding. New York.
4. American National Standards Institute/International Standards Organization  
ANSI X3.122/ISO 8632. Information Processing Systems - Computer Graphics  
Metafile for the storage and transfer of picture information.  
Part 4: Clear text encoding. New York
5. AMF Logic Sciences, Inc.  
HSR -11B Manual.  
Logic Sciences, Inc. Houston Texas. USA. 1982.
6. Dobrin, Milton B  
Introducción a la prospección geofísica.  
Editorial Omega. Segunda edición Barcelona Esp. 1975.
7. Del Valle Toledo, Enrique.  
Apúntes de introducción a los métodos geofísicos de exploración.  
División de Ingeniería en Ciencias de la Tierra. Depto. de Geofísica.  
Facultad de Ingeniería UNAM. 1987.
8. Pressman, Roger S.  
Ingeniería del software. Un enfoque práctico.  
Mc Graw-Hill Segunda edición. México D.F 1990.

9. Kernighan, Brian W., Ritchie, Dennis M  
El lenguaje de programación C.  
Segunda edición Prentice Hall. N.J. 1989.
10. Kernighan, Brian W., Pike, Rob.  
El entorno de programación UNIX.  
Prentice Hall N.J. 1984