

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS



*SIMULACIÓN INTERACTIVA DE MODELOS GEOFÍSICOS*

*CON UNA INTERFAZ TIPO X WINDOW*

Tesis Profesional

Que para obtener el Título de:

**LICENCIADO EN CIENCIAS COMPUTACIONALES**

Presenta:

***MARÍA DE JESÚS CERVANTES CERDA***

Junio, 1997.

Ensenada, B.C. México.

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**

**FACULTAD DE CIENCIAS**

***SIMULACIÓN INTERACTIVA DE MODELOS GEOFÍSICOS***

***CON UNA INTERFAZ TIPO X WINDOW***

**TESIS PROFESIONAL**

**QUE PRESENTA:**

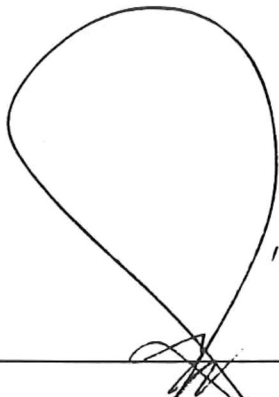
***MARÍA DE JESÚS CERVANTES CERDA***

**APROBADO POR:**



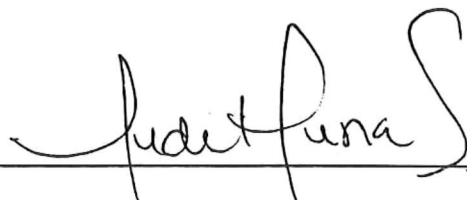
---

**DR. JUAN GARCÍA ABDESLEM**  
**Presidente del jurado**



---

**FÍS. FRANCISCO JUÁREZ GARCÍA**  
**SECRETARIO**



---

**OC. JUDITH ISABEL LUNA SERRANO**  
**1er. VOCAL**

## DEDICATORIA

A mi mamá: Yolanda Cerda Castañeda  
por su amor y apoyo en todos los aspectos de mi vida, por darme la libertad  
de ser y elegir en todo momento. Te admiro por tu gran corazón.

A mi hermana Esther Yolanda Rodríguez Cerda  
por ser un ejemplo como profesional, estudiante y madre para seguir  
superándome. Te quiero mucho.

A mis sobrinos Rubén Eduardo y Josué David por ser con sus juegos y risas  
un aliciente para mi vida.

## AGRADECIMIENTOS

A mi director de tesis Dr. Juan García Abdeslem por su paciencia y apoyo.

A mi sinodal Oc. Judith Isabel Luna Serrano por las aportaciones, asesorías y sugerencias en la realización de mi tesis. Por ser mi amiga en todo momento.

A mi sinodal Fís. Francisco Juárez García por su buena disposición y colaboración en mi tesis.

A la Universidad Autónoma de Baja California, en especial a la Facultad de Ciencias por permitir desarrollarme como estudiante y ahora en mi carrera profesional.

Al Centro de Investigaciones y de Educación Superior de Ensenada, de manera especial al Departamento de Ciencias de la Tierra donde realicé mi trabajo de tesis y me impulsaron hasta llegar al término de ésta.

A M.C. Gloria Rubí Vázquez, M.C. Irma Rivera Garibaldi y M.C. Miguel Carrillo por su apoyo en el tiempo que he estado trabajando en la U.A.B.C.

A Carlos Famoza por su apoyo técnico y asesorías en la realización de este trabajo.

A mis compañeros de trabajo, especialmente a la L.C.C. Martha Irene Estrada por su apoyo incondicional para la finalización de este trabajo y por tu amistad, cariño y compañerismo en estos años.




A mis tíos Lorenzo López C. y Jesús Cervantes C. por el gran cariño que me han brindado .


A Zelmira Cañedo , Emilia Alaniz y Eduardo Robles por la amistad que mantenemos a pesar de la distancia.

A la familia Arvizu Coronado porque comparten con mi familia como si fuéramos una sola familia. En especial a mi cuñado Rubén Arvizu Coronado por el apoyo que me ha brindado para llegar a esta meta.

A la Profra. Leonor Ramírez Yañez y fam. por su amistad constante y compartir los momentos de dicha y momentos difíciles con mi familia y conmigo.

## AGRADECIMIENTOS


A Martha Carrasco , Magui Cervantes , y Mayra Contreras  por su amistad y aguantarme mis malos ratos y disfrutar los super excelentes buenos ratos entre el “vamos a platicar”, desayunos, desveladas y películas, ah! y por echarme porras mientras finalizaba mi tesis. Chicas, definitivamente las quiero mucho.

A la 6a. generación de Lic. en Ciencias Computacionales  por su compañerismo y apoyarnos unos a otros, especialmente a Ángel Dueñas, Edgar Manzanarez, Víctor Torres, Alejandro León, Fco. Javier Gaynor y José L. Godoy.

A Gustavo Barbosa por tu amistad sincera y ayuda.

A Juanita y Ricardo Luzanilla por brindarme su amistad cariñosa y su confianza.

A la Sra. Dominga Agúndez por sus pláticas.

A mis amigos de Amistad Familiar por su cariño sincero fomentado en la palabra de vida y que crece día con día, en especial  a los miembros de la alabanza Gaby, Sonia, Pech, Leony, Waldo, Josué y Beto por los tiempos compartidos. A Josué Alvarez y fam., a Tere y Juan Fitzner, a Sra. Hortensia y Don Octavio Jiménez, a Lily y Francisco Palomino, a la Sra. Anita y a Rosalba.

A Sara Vázquez que me invitó a irnos a estudiar a Ensenada, y aquí estamos todavía.

A Olimpia Rivera por saber que cuento contigo y eres especial para mi.

A Víctor Plata, por ser amable, sincero y amigo sobretodo.

A Eleazar Canale por ser mi amigo, aunque estés lejos.

A Hugo Silva porque has sido mi amigo incondicional.

A Tatiana, Ricardo, Rigo, Eduardo, Jesús, Ramón y Milanez por colaborar en las labores diarias del aula equipada, y que ha sido de gran apoyo para mi.

A mis hermanos Lucy, Adrián, René y Juan por ser buenos chicos y el cariño que me brindan cuando los veo.

A los que escapan de mi pensamiento y no pensé que se me olvidarían.

## AGRADECIMIENTOS

A ti, bendita Trinidad (Padre , Hijo y Espíritu Santo) que desde que te conocí de verdad, no de oídas diría Job (42:5), sabes que eres lo primero para mi,  
mi Dios , mi Salvador,

mi Redentor, Protector,  
mi Consolador, Proveedor,

gracias Padre, porque lo mejor es estar en tu presencia y saber que me amas tanto que diste a tu hijo unigénito Jesucristo por mi, ahora sé que puedo hablar contigo, estar contigo, y tener una relación personal contigo como amigos, gracias al Espíritu Consolador (*“No temas, porque yo estoy contigo; no desmayes, porque yo soy tu Dios que te esfuerza; siempre te ayudaré, siempre te sustentaré con la diestra de mi justicia (Isaias 41:10)”*)

Mil gracias! Señor por tu palabra ( la Biblia) en la que nos das tu plan de amor desde el principio para nosotros. Tanto mas podría decir de tus maravillas (Salmo 139:14) que no terminaría. Pero en fin, no me queda mas que esperar todo lo que nos has preparado. ok? (*“Cosas que ojo no vio, ni oído oyó, ni han subido a corazón de hombre, son las que Dios ha preparado para los que le aman.” 1a. de Corintios 2:9*)

Tu hija: Mary Cervantes

Resumen de la tesis de María de Jesús Cervantes Cerda como requisito parcial para la obtención de la licenciatura en CIENCIAS COMPUTACIONALES. Ensenada, Baja California. México. Junio 1997.

## SIMULACIÓN INTERACTIVA DE MODELOS GEOFÍSICOS CON UNA INTERFAZ TIPO X WINDOW



Resumen Aprobado:

---

Dr. Juan García Abdeslem  
Director de tesis

En este trabajo se utiliza la metodología de Jackson para analizar dos modelos geofísicos y para diseñar una plataforma computacional que permite realizar cambios en los parámetros de ambos modelos, y visualizar, simultáneamente, en una interfaz gráfica, el cambio en la respuesta de los modelos. La plataforma computacional fue desarrollada utilizando el sistema X Window y el lenguaje C.

Los modelos geofísicos utilizados nos permiten simular el efecto gravimétrico producido por una esfera, cuya densidad es constante, y se encuentra sepultada en el interior de la tierra, y el efecto gravimétrico producido por un prisma bidimensional, cuya densidad varía con la profundidad. El modelo de la esfera puede ser utilizado para la interpretación de anomalías gravimétricas producidas por domos salinos y cavidades. El modelo del prisma es útil en la interpretación de anomalías gravimétricas debidas a cuencas sedimentarias, en donde la densidad de los sedimentos cambia con la profundidad, así como para el estudio de fallas geológicas.

---

<b>ÍNDICE</b>		
	CONTENIDO	PÁGINA
CAPÍTULO I.	INTRODUCCIÓN	1
CAPÍTULO II.	ANTECEDENTES	3
CAPÍTULO III.	OBJETIVOS	5
CAPÍTULO IV.	ANOMALÍAS GRAVIMÉTRICAS	6
	IV.1. SOLUCIÓN AL PROBLEMA DIRECTO	12
	IV.2. ATRACCIÓN GRAVITACIONAL	
	PRODUCIDA POR UNA ESFERA	16
	IV.3. RESTRICCIONES DEL SISTEMA	17
	IV.4. ATRACCIÓN GRAVITACIONAL	
	PRODUCIDA POR UN PRISMA	20
CAPÍTULO V.	METODOLOGÍA PARA LA SIMULACIÓN	29
	V.1. INFRAESTRUCTURA DE	
	SOFTWARE Y EQUIPO DE CÓMPUTO	29
	V.2. METODOLOGÍA DE JACKSON	30
CAPÍTULO VI.	RESULTADOS	32
	VI.1. ANÁLISIS DEL SISTEMA GEOFÍSICO	32
	VI.1.1. ANÁLISIS DE ENTIDADES Y ACCIONES	
	QUE AFECTAN LAS ENTIDADES	32

---

---

VI.1.2. DIAGRAMA DE ESTRUCTURA DE ENTIDADES	32
VI.1.3. MODELO INICIAL DEL ANÁLISIS	34
VI.2. DISEÑO DEL SISTEMA GEOFÍSICO	35
VI.2.1. FUNCIONES	35
VI.2.2. SINCRONIZACIÓN	35
VI.2.3. IMPLEMENTACIÓN	35
VI.2.3.1. PSEUDOCÓDIGO PARA LA ANOMALÍA DE LA ESFERA	36
VI.2.3.2. PSEUDOCÓDIGO PARA LA ANOMALÍA DEL PRISMA	38
VI.2.4. INTERFAZ DE ENTRADA Y SALIDA DE DATOS	40
VI.2.5. DICCIONARIO DE DATOS	41
VI.2.5.1. LISTA DE ELEMENTOS	43
VI.2.5.2. LISTA DE PROCESOS PARA LA ESFERA	44
CAPÍTULO VII. DISCUSIÓN	49
CAPÍTULO VIII. CONCLUSIONES	51
LITERATURA CITADA	53
APÉNDICE A	55
PROGRAMAS FUENTES Y BIBLIOTECAS	55

---

---

## ÍNDICE

### LISTA DE FIGURAS

		PÁGINA
FIGURA 1.	GRAVEDAD TEÓRICA	8
FIGURA 2.	DISTRIBUCIÓN DE MASA	15
FIGURA 3.	ANOMALÍA DE UNA ESFERA 3-D	18
FIGURA 4.	PERFIL DE LA ANOMALÍA DE UNA ESFERA	19
FIGURA 5.	GRÁFICA DE LA DENSIDAD	26
FIGURA 6.	MAPA DE CONTORNOS	27
FIGURA 7.	DIAGRAMA DE ESTRUCTURAS DE DATOS	33
FIGURA 8.	MODELO INICIAL DEL ANÁLISIS	34
FIGURA 9.	DISEÑO DE INTERFAZ GRÁFICA	42
FIGURA 10.	INTERFAZ GRÁFICA PARA UNA ESFERA	46
FIGURA 11.	INTERFAZ GRÁFICA PARA UN PRISMA	47

---

## ÍNDICE

### LISTA DE TABLAS

	PÁGINA
TABLA I. VALORES DE LAS ABSCISAS Y PESOS UTILIZADOS EN EL MÉTODO DE CUADRATURA DE GAUSS-LEGENDRE	24

## I. INTRODUCCIÓN

El avance en las Ciencias Computacionales está correlacionado con avances en otras áreas de la ciencia, en donde su participación ha sido de gran ayuda para aumentar la rapidez en procesos que, en otro tiempo, se hubiera llevado años realizar, o que bien jamás se podrían haber realizado.

Una de las tareas características en el departamento de Geofísica Aplicada del CICESE consiste en el desarrollo de modelos matemáticos para interpretar diversos tipos de datos geofísicos, a partir de los cuales se infieren las propiedades físicas y la geometría de estructuras geológicas de interés. No obstante la utilidad intrínseca de dichos modelos, sería conveniente poder visualizar los resultados de este tipo de investigaciones, en el monitor de una computadora, en forma rápida y amigable; esto sería de gran ayuda, particularmente en docencia. Existen productos en el mercado que permiten la simulación, o modelado, de sistemas geofísicos. Sin embargo, desde una perspectiva a mediano plazo es deseable contar con gente capacitada para el desarrollo de productos similares o mejores.

En exploración geofísica, se realizan diversos tipos de mediciones sobre la superficie de la tierra, a partir de las cuales se trata de inferir alguna propiedad física de los materiales que constituyen el subsuelo, por ejemplo: densidad de masa, resistividad eléctrica o velocidad de propagación de ondas sísmicas, entre otras.

Uno de los métodos de prospección geofísica, empleado comúnmente para el estudio del interior de la tierra, es el método gravimétrico (Telford, et al. , 1976). En este método se realizan mediciones de la componente vertical del campo de gravedad terrestre ( $g$ ), con el objetivo de inferir la presencia de cambios en la densidad de las rocas y materiales del subsuelo. Para ello se utilizan gravímetros, capaces de medir la gravedad con una precisión del orden de una parte en  $10^6$ . La unidad de medición empleada comúnmente en trabajos de exploración es el miligal ( $1 \text{ mGal} = 10^{-5} \text{ m/s}^2$ ). En el trabajo de prospección, las mediciones de  $g$  se realizan a lo largo de transectos, con estaciones separadas entre 50 y 500 metros, dependiendo del objetivo geológico, o con estaciones distribuidas regular o irregularmente en el área de estudio.

La variación espacial de  $g$ , una vez que son considerados una serie de efectos conocidos, que afectan nuestra medición, pueden ser interpretadas como debidas a cambios laterales en la densidad de los materiales que constituyen el interior terrestre. Esto nos permite inferir la presencia de yacimientos minerales o el estudio de cuencas sedimentarias con objetivos petroleros o hidrológicos.

## II. ANTECEDENTES

Existen en la naturaleza una gran diversidad de fenómenos que impactan en la actividad económica y cultural del hombre; uno de los objetivos de la ciencia es el de poder entenderlos y describirlos mediante leyes físicas. Una vez que se ha entendido un fenómeno y se cuenta con una ley que lo describe es posible simularlo. La simulación permite imitar a la naturaleza ayudando a entenderla y dominarla ().

De acuerdo con Tremblay y Bunt (1988) la simulación implica la construcción de modelos en forma de programas de computación, modelos que se usan en el estudio de algunos sistemas o fenómenos. El modelo proporciona un objeto más conveniente de experimentación que el sistema real, puesto que es más fácil de tenerlo bajo control (Tremblay y Bunt, 1988).

El desarrollo científico y tecnológico acelerado que se vive en nuestros días es, en gran medida, debido a que las computadoras han permitido simular y estudiar un sinnúmero de fenómenos. Con la llegada de la computación (Coss, 1982) se han desarrollado una gran cantidad de herramientas analíticas que han tenido un profundo impacto en el campo científico. Una de estas herramientas es la simulación, cuyos usos y aplicaciones se han extendido en los últimos años (Law, 1991; Coss, 1982).

Mediante la simulación, a través de un modelo computarizado de un sistema o proceso, es posible conducir experimentos con el propósito de entender el comportamiento de un sistema o para evaluar varias de las estrategias posibles con las cuales se puede operar el sistema (Shannon, 1988). Estos experimentos involucran

modelos matemáticos y lógicos que nos permiten describir el comportamiento de un sistema.

El tema de la tesis es la simulación computacional de modelos geofísicos con una interfaz tipo X Window. El lenguaje de programación elegido para desarrollar el programa es el lenguaje C estándar, sin embargo, debido a que el lenguaje C en una plataforma UNIX, no nos permite la salida gráfica requerida, el lenguaje C se utiliza en conjunto con el sistema X Window, el cual si nos permite una interfaz gráfica de alta calidad.

En este trabajo se aborda el análisis de un sistema de tipo geofísico. La simulación se realiza con el propósito de entender el comportamiento del sistema ante el cambio de sus parámetros, sin necesidad de experimentar directamente sobre un sistema físico equivalente. Los experimentos que se pueden llevar a cabo no requieren del factor tiempo como una variable importante. Los valores de las variables que componen este modelo pueden cambiar instantáneamente en puntos separados de tiempo. Las variables del modelo se definen de manera explícita y son independientes del tiempo, es decir, este modelo corresponde a un sistema del tipo determinístico y discreto (Law, 1991).

### **III. OBJETIVOS**

Los objetivos de este trabajo consisten en: a) Diseñar e implementar una aplicación computacional que permita simular un sistema geofísico; y b) Desarrollar una interfaz gráfica y amigable para el usuario que permita visualizar la respuesta del sistema cuando cambian sus parámetros.

#### **Descripción del sistema**

El sistema geofísico que se analiza corresponde a la atracción gravitacional producida por una masa esférica, cuya densidad es constante, y por un prisma bidimensional en donde la densidad varía con la profundidad de acuerdo a una función dada. En el caso de la esfera, sus parámetros libres son: el radio de la esfera, la profundidad a su centro de masa y su densidad. En el caso del prisma, sus parámetros son: el espesor y los coeficientes de la función que describe la densidad. Los cambios en los parámetros del sistema se realizarán de manera amigable, directamente desde el monitor de una computadora, con el uso de menús y del ratón.

El sistema X Window permite la creación de ventanas y ambientes gráficos dentro de éstas, las cuales se utilizan para representar todas las variables del sistema, el modelo y su respuesta gravimétrica y la presentación de menús.

#### IV. ANOMALÍAS GRAVIMÉTRICAS

Enseguida se presenta una breve introducción acerca del significado de una anomalía gravimétrica. En prospección gravimétrica se realizan mediciones de la componente vertical del campo de gravedad (en la superficie terrestre), con el propósito de localizar masas anómalas en el subsuelo. Para su análisis, las mediciones gravimétricas se simplifican hasta obtener anomalías de Bouguer, las cuales se interpretan como debidas a cambios laterales en la densidad de las rocas del subsuelo, usualmente interpretados como cambios en el tipo de rocas. Este método de exploración ha permitido detectar estructuras geológicas de interés económico, como pueden ser yacimientos minerales y domos salinos, o para inferir la geometría y extensión a profundidad de estructuras geológicas de carácter regional, y para el estudio de acuíferos y campos petroleros en zonas sedimentarias.

De una manera muy simplificada, la gravedad observada ( $gO$ ) puede ser expresada como la suma del efecto gravimétrico debido a causas predecibles ( $gP$ ) más el efecto gravimétrico debido a la estructura geológica ( $gG$ ) que se estudia; esto es:

$$gO = gP + gG \quad (1)$$

Entre las causas predecibles, las cuales podemos evaluar, está el efecto gravimétrico producido por la distribución interna de la masa, forma y rotación de la Tierra. Este efecto es conocido como la gravedad normal o gravedad teórica ( $gT$ ), la cual es descrita como una función de la latitud geográfica, sobre el esferoide de referencia que mejor describe la forma del planeta. El campo de gravedad normal ha sido objeto de estudio desde tiempos de Newton. En 1967 la Asociación Internacional de Geodesia adoptó y recomendó el uso de una expresión para describir la gravedad normal, basada en un elipsoide cuya elipticidad es  $e = c / (c - a) = 1/298.247$ , en donde  $c$  y  $a$  son respectivamente, el radio polar y ecuatorial de la Tierra. La fórmula internacional de la gravedad de 1967 está dada por:

$$gT = gE \left( 1 + \alpha \operatorname{sen}^2 \phi - \beta \operatorname{sen}^2 2\phi \right) , \quad (2)$$

en donde  $gE = 978,031.85$  mGal, corresponde al valor de la gravedad teórica, al nivel del mar, en el Ecuador terrestre;  $\alpha = 0.0053024$  y  $\beta = 0.00000587$  son dos constantes que dependen de la distribución de masa en el interior del planeta, de su forma y de su rotación;  $\phi$  denota latitud geográfica. Una descripción gráfica de la gravedad teórica descrita por la Fórmula Internacional de 1967, desde el Ecuador al Polo Norte, se presenta en la Figura 1.

Considerando que, comúnmente las mediciones de gravedad se realizan en sitios cuya elevación está por encima del esferoide de referencia, el cual en promedio, coincide con la superficie libre del mar (excluyendo el efecto de mareas y corrientes), es necesario

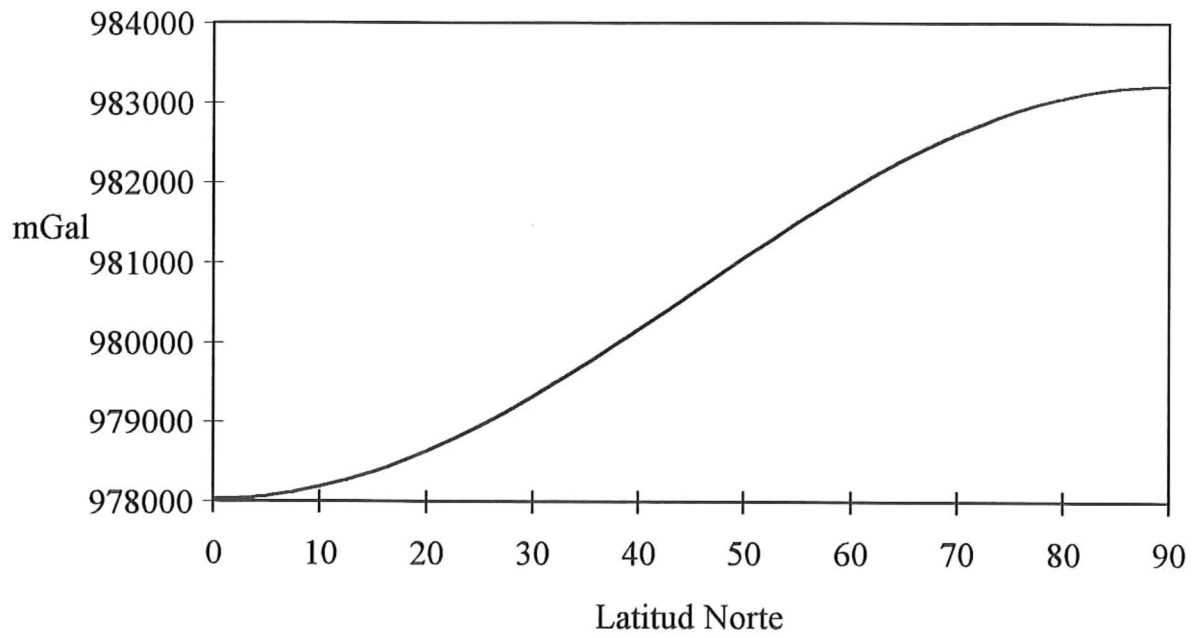


Figura 1. Variación de la gravedad normal como función de la latitud geográfica.

tomar en cuenta el gradiente vertical del campo de gravedad normal. El cual, debido a que no toma en cuenta la atracción gravitacional ejercida por la masa situada entre el sitio de medición y el esferoide de referencia, es conocido como el efecto de aire libre. El efecto de aire libre a una elevación  $h$  sobre el nivel de referencia se obtiene de la manera siguiente. La gravedad ( $g_0$ ) al nivel del mar, sobre una Tierra esférica está dada por:

$$g_0 = G \frac{M}{R^2} \quad , \quad (3)$$

en donde  $G$  es la constante de la gravitación de Newton,  $M$  es la masa de la Tierra y  $R$  su radio; la gravedad correspondiente a la elevación  $h$  es:

$$gh = G \frac{M}{(R+h)^2} = \frac{GM}{R^2} \frac{1}{\left(1+h/R\right)^2} \quad (4)$$

Expresando  $gh$  en términos de  $g_0$ , y expandiendo la expresión en una serie binomial, se obtiene que:

$$gh = g_0 \left[ 1 - 2\left(\frac{h}{R}\right) + 3\left(\frac{h}{R}\right)^2 + \dots \right] \quad , \quad (5)$$

a partir de la cual, sustituyendo el radio terrestre, se obtiene el efecto de aire libre,

$$gAL = 0.3086 h - 0.073 H^2 , \quad (6)$$

en el cual, si la elevación del sitio de medición ( $h$ ) se da en  $m$  y ( $H$ ) en  $km$ , las unidades de  $gAL$  se obtienen en mGals.

La atracción gravitacional producida por la masa situada entre el nivel de referencia (nivel del mar) y el de la estación, se considera en la corrección de Bouguer. El efecto de Bouguer ( $gB$ ) está dado por :

$$gB = 41.93 \rho H , \quad (7)$$

en donde  $\rho$  denota la densidad del material situado entre la superficie del mar y la elevación ( $H$ ) del sitio de medición. Cuando la densidad de Bouguer se expresa en  $g/cm^3$  y la elevación en  $km$ ,  $gB$  tiene unidades de mGal.

Todas las consideraciones anteriores nos llevan a aislar la anomalía gravimétrica residual, producida por nuestro objetivo geológico, de la siguiente forma:

$$gG = gO - gP . \quad (8)$$

Dependiendo del tipo de causas predecibles que estemos tomando en cuenta será el nombre de la anomalía que se obtenga. Por ejemplo, para mediciones efectuadas al nivel del mar únicamente necesitamos eliminar el efecto de la gravedad teórica, con lo cual, la anomalía gravimétrica está dada por:

$$\Delta g(AL) = gO - gT . \quad (9)$$

Cuando efectuamos mediciones por encima, o por debajo, del nivel del mar necesitamos compensar por el efecto de aire libre, con lo cual obtenemos la anomalía de aire libre

$$\Delta g(AL) = gO - gT + gAL . \quad (10)$$

Incluyendo el efecto gravimétrico debido a la masa situada entre el nivel de referencia y el sitio de medición, obtenemos lo que se conoce como la anomalía de Bouguer:

$$\Delta g(B) = gO - gT + gAL + gB . \quad (11)$$

Una vez que se ha calculado la anomalía gravimétrica residual, se procede a su interpretación. Para ello, la primera fase del proceso de interpretación consiste en representar en forma gráfica la anomalía gravimétrica residual. Esto, dependiendo del tipo de estudio, puede hacerse en forma de perfiles o por medio de mapas de isolíneas de igual

valor de anomalía. Ello permite realizar una inspección visual de los datos. En esta primera etapa de la interpretación cualitativa, buscamos la presencia de valores extremos (v.g., mínimos y máximos) que pongan en evidencia un cambio drástico en la densidad de las rocas del subsuelo o algún tipo de tendencia lineal o curvilineamientos.

Habiendo definido el objetivo geológico, en una segunda etapa del proceso de interpretación, usualmente conocida como cuantitativa, se buscará determinar la geometría y la densidad de la masa anómala que causa la anomalía. Este proceso se puede realizar siguiendo un procedimiento de prueba y error, o en forma automática siguiendo algún criterio de optimización, hasta lograr, en ambos casos, que el efecto gravimétrico de la estructura propuesta coincida, lo mejor posible, con la anomalía gravimétrica residual observada, cuidando siempre que la interpretación tenga sentido geológico.

En ambos procedimientos de interpretación se requiere determinar, en forma analítica, la expresión matemática que describe el efecto gravimétrico debido a formas geométricas simples, como pueden ser: esferas, cilindros, prismas o poliedros. Esto se conoce como la solución al problema directo. El conocimiento de la respuesta gravimétrica debida a dichas formas geométricas permitirá, eventualmente, poder diagnosticar en forma adecuada el modelo más apropiado para interpretar la anomalía observada.

#### **IV.1. Solución al problema directo**

En esta sección se describen los principios físicos que gobiernan la atracción gravitacional así como su aplicación en cálculo del efecto gravitacional producido por una esfera y por un prisma. La fuerza de atracción entre dos masas puntuales o esféricas,  $m$  y  $m'$ , separadas entre si una distancia  $r$ , es descrita por la Ley de la Gravitación de Newton: La fuerza de atracción entre dos masas ( $m'$  y  $m$ ) es proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia ( $r$ ) que las separa,

$$\mathbf{F} = G \frac{m m'}{r^2} \hat{\mathbf{r}} \quad , \quad (12)$$

la fuerza ( $\mathbf{F}$ ) actúa en la dirección definida por el vector unitario  $\hat{\mathbf{r}}$ , dirigido desde la posición de la masa  $m'$ , hacia la posición de la masa  $m$ . La fuerza ejercida sobre la masa  $m$  por la masa  $m'$  es idéntica en magnitud a la fuerza ejercida por  $m$  en  $m'$  pero con sentido opuesto.

La aceleración se define como la fuerza por unidad de masa. Por lo tanto, la aceleración de la masa  $m'$  hacia la masa  $m$  es:

$$\mathbf{g} = \frac{\mathbf{F}}{m'} = G \frac{m}{r^2} \hat{\mathbf{r}} \quad (13)$$

Esta ecuación describe en forma completa el campo de gravedad producido alrededor de la masa  $m$ .

Si consideramos ahora un cuerpo de forma arbitraria, constituido por una colección de  $N$  masas puntuales, tenemos que el campo gravitacional que genera a su alrededor en la posición  $r'$ , estará dado por la superposición de los campos gravitacionales generados por la colección de masas puntuales; esto es:

$$\mathbf{g}(r') = G \sum_{i=1}^N \frac{m_i}{(r' - r_i)^2} \hat{\mathbf{r}}_i \quad (14)$$

Podemos generalizar la expresión anterior para calcular la componente vertical del campo gravitacional (en la dirección de  $\hat{\mathbf{z}}$ ) producido por una distribución continua de masa contenida en el volumen  $V$ ; esto es:

$$g(r') = G \int_V d\mathbf{r}'^3 \frac{\mathbf{r} \cdot \hat{\mathbf{z}}}{(r' - \mathbf{r})^{3/2}} \rho(\mathbf{r}') \quad , \quad (15)$$

en donde la masa ha sido sustituida por el producto entre la densidad ( $\rho$ ) del material y el volumen que lo contiene (Figura 2).

Comúnmente, la evaluación de la anterior integral se realiza escogiendo un sistema de coordenadas apropiado a la geometría de la masa anómala. En la siguiente sección se presenta la formulación de la solución al problema directo para dos casos

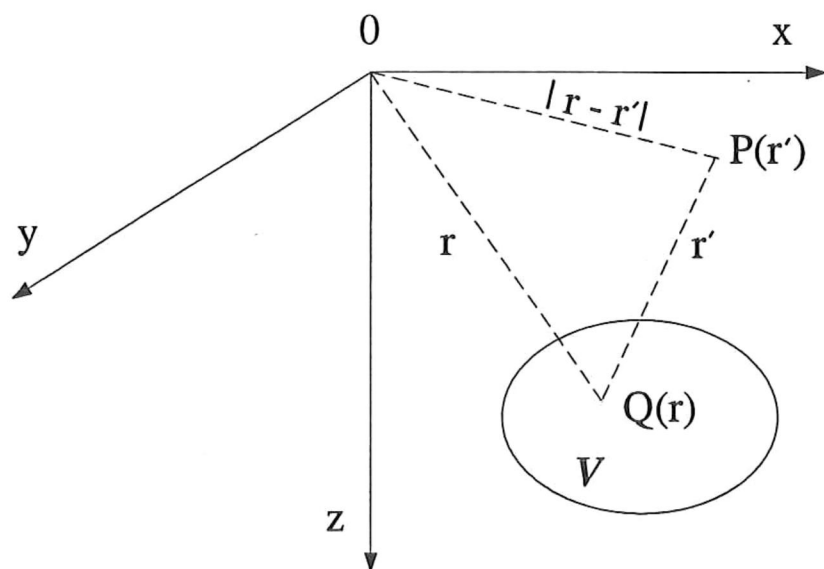


Figura 2. Distribución continua de masa contenida en el volumen  $V$ .

particulares: cuando la geometría del cuerpo anómalo corresponde a una esfera cuya densidad es constante; y cuando la geometría del cuerpo corresponde a la de un prisma vertical bidimensional (v.g., en donde una de sus dimensiones se extiende al infinito).

#### IV.2. Atracción gravitacional producida por una esfera

La solución a este problema se facilita considerando que la masa de la esfera está concentrada en su centro de masa. Por lo tanto sustituyendo  $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}|$  en la ecuación (13) podemos escribir la componente vertical del campo de gravedad, en la dirección del vector unitario  $\hat{\mathbf{z}}$ , como

$$g_z = G m \frac{\mathbf{r} \cdot \hat{\mathbf{z}}}{r^{3/2}} \quad , \quad (16)$$

Sustituyendo la masa de la esfera por el producto de su volumen por su densidad, y realizando el producto punto, se obtiene que:

$$g_z(x', y', z') = \frac{G4\pi R^3 \rho}{3} \frac{(z - z')}{\left[ (x - x')^2 + (y - y')^2 + (z - z')^2 \right]^{3/2}} \quad , \quad (17)$$

en donde  $g_z$  representa la anomalía gravimétrica producida por la esfera, calculada en el punto de coordenadas  $(x', y', z')$ ;  $G$  es la constante gravitacional;  $\rho$  es la densidad de la

esfera; y las coordenadas de un punto material de la esfera se denotan por  $(x, y, z)$ . En la Figura 3, se muestra la anomalía gravimétrica debida a una esfera; calculada con la ecuación (17).

### IV.3. Restricciones del sistema

Dada la simetría de problema, únicamente se consideró la anomalía gravimétrica observada en la dirección del eje  $x'$ , en  $y' = 0$  y en el plano  $z' = 0$ . Adicionalmente, el centro de masa de la esfera estará localizado en el punto de coordenadas  $(0, 0, z)$ . Con estas restricciones, la ecuación (16) se simplifica de la siguiente manera:

$$g(x') = \frac{G4\pi R^3 \rho}{3} \frac{z}{[x'^2 + z^2]^{3/2}} \quad (18)$$

De esta forma, el centro de la esfera siempre estará fijo en  $x = 0$  pudiendo variar su posición sobre el eje  $z$ . Adicionalmente, la densidad en el modelo esférico se restringió al rango de valores típicos de las rocas de la corteza terrestre: de  $-3$  a  $3 \text{ g/cm}^3$ . El rango de valores para designar la posición del observador se fijó entre  $-50$  y  $50 \text{ m}$ .

Después de estas simplificaciones, las variables activas del problema son: el radio de la esfera, la profundidad a su centro de masa, su densidad y la posición del observador. En la Figura 4, se muestra el perfil de la anomalía gravimétrica producida por una esfera, calculada en la ecuación (18).

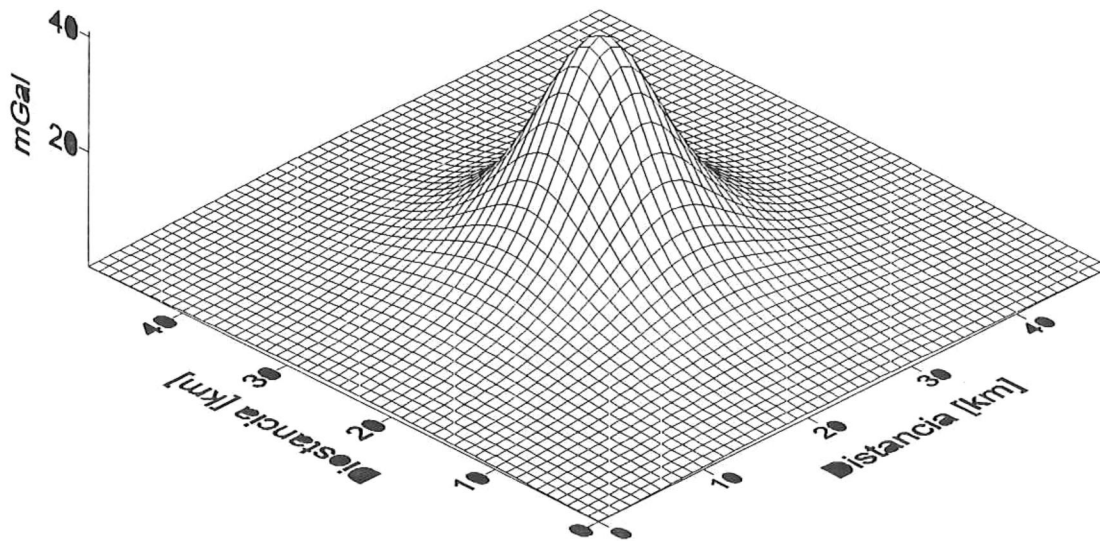


Figura 3. Anomalía gravimétrica producida por una esfera. La esfera tiene un radio de 5 km, su densidad es de  $1000 \text{ kg/m}^3$  y su centro de masa está localizado a 8 km de profundidad.

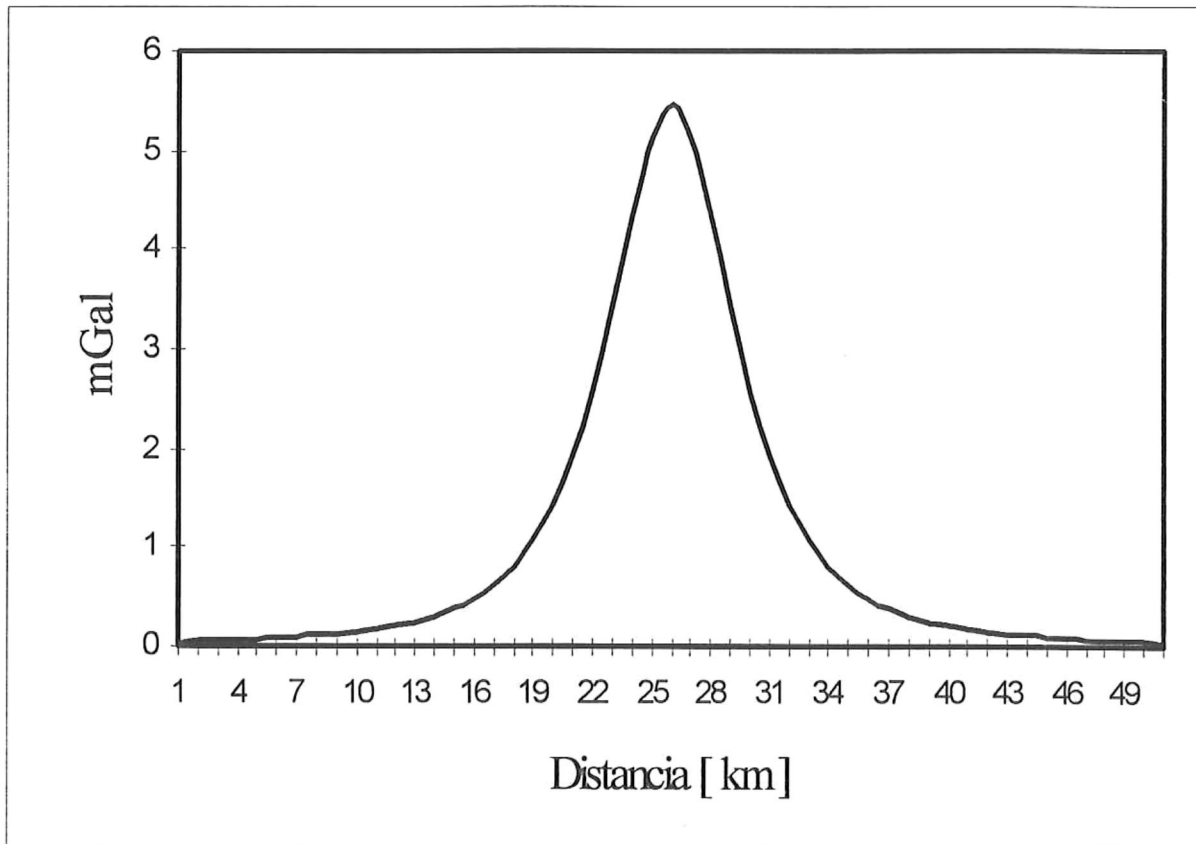


Figura 4. Perfil de la anomalía gravimétrica producida por una esfera.

#### **IV.4. Atracción gravitacional producida por un prisma**

En la interpretación de anomalías gravimétricas, el uso de modelos geofísicos para simular estructuras geológicas de dos y tres dimensiones donde la densidad varía con la profundidad, proporciona algunas ventajas computacionales con respecto a esquemas de modelado más tradicionales, en donde se utilizan cuerpos cuya densidad es constante. Modelos gravimétricos, cuya densidad es no uniforme, son útiles en escenarios geológicos complejos, particularmente en el análisis de datos gravimétricos en cuencas sedimentarias, donde, normalmente, la densidad de los sedimentos que rellenan la cuenca, aumenta con la profundidad de sepultamiento.

Revisando la literatura reciente se encuentra que se ha puesto especial interés al caso en el cual la densidad se incrementa con la profundidad en forma exponencial. Este es un caso particular, el cual ocurre en una cuenca sedimentaria cuando el proceso diagenético más importante es la compactación diferencial simple de los sedimentos (Athy, 1930) y ha sido tratado por Cordell (1973) y por Chai y Hinze (1988), quienes han calculado el efecto gravimétrico producido por un prisma rectangular. El caso de un incremento lineal de la densidad con la profundidad ha sido abordado por Murthy y Rao (1979) y más recientemente Rao (1990) ha derivado una expresión analítica para el cálculo del efecto gravimétrico producido por un trapecio asimétrico cuya densidad varía con la profundidad de acuerdo con un polinomio de segundo grado.

Además del proceso de compactación diferencial simple, en la evolución geológica de una cuenca sedimentaria ocurren otros procesos geológicos, tales como: la depositación de

sedimentos sobre un relieve preexistente y estratificación no uniforme, cambios laterales del medio ambiente de depositación que se reflejan en cambios de faces, cementación química y perturbaciones de origen tectónico. Debido a ello, se requiere de un método capaz de acomodar variaciones arbitrarias de la densidad con la profundidad.

En trabajos recientes García-Abdeslem (1992; 1996) ha desarrollado un método para calcular el efecto gravimétrico debido a un cuerpo prismático cuya densidad varía con la profundidad de acuerdo con una función arbitraria. Dicho método combina métodos de integración analíticos y numéricos para los casos de estructuras en dos y tres dimensiones. Aparte de la simplicidad computacional del método, una de sus ventajas sobre métodos anteriores es que cualesquier función bien comportada, que excluya singularidades en su dominio de definición, puede ser utilizada para representar la densidad como una función de la profundidad. Enseguida se describe la solución al problema directo.

En un sistema derecho de coordenadas cartesianas la componente vertical del campo de gravedad  $g$  en un punto  $P(x_0, z_0)$  producido por un prisma bidimensional de lados verticales, cuya densidad  $\rho(z)$  varía únicamente con la profundidad, está dado por:

$$g(x_0, z_0) = 2G \iint_{XZ} dx dz \frac{(z - z_0) \rho(z)}{(x - x_0)^2 + (z - z_0)^2} , \quad (19)$$

donde  $G$  es la constante de la gravitacional de Newton;  $(x, z)$  son las coordenadas materiales de la fuente (el prisma),  $x$  es la variable espacial perpendicular al rumbo del cuerpo y  $z$  es la

profundidad, considerada positiva hacia abajo. Realizando la integración de la ecuación (19) sobre la variable  $x$  se obtiene:

$$g(x_0, z_0) = 2G \int_z dz \sigma(z) \arctan[(x - x_0) / (z - z_0)] \Big|_{x_1}^{x_2}, \quad (20)$$

en donde los límites ( $X_1, X_2$ ) denotan los bordes del prisma a lo largo del eje  $x$ . La integración de la ecuación (20) sobre la variable  $z$  no ha podido ser resuelta en forma analítica, debido a lo cual se utiliza el método de integración conocido como Gauss-Legendre (Atkinson, 1989) para el caso de un intervalo arbitrario de integración, con lo cual se reescribe la ecuación (20) de la forma siguiente:

$$g(x_0, z_0) = G (Z_2 - Z_1) \sum_{i=1}^n w_i f(\zeta_i) + R_n, \quad (21)$$

en donde  $Z_1$  y  $Z_2$  son los límites de integración de la variable  $z$ , los cuales corresponden, respectivamente, a la parte superior e inferior del prisma,  $R_n$  representa el error en la integración;  $f(\zeta)$  representa el integrando de la ecuación (20); y  $w_i$  representa valores ponderados, que se expresan en función de la primera derivada de la función de Legendre de la primera clase como:

$$w_i = \frac{2}{(1 - \kappa_i^2) [P_n(\kappa_i)]^2} , \quad (22)$$

en donde  $\kappa_i$  es el  $i$ -ésimo cero de la función de Legendre de la primera clase. El integrando  $f(\zeta)$  en la ecuación (21) será evaluado en  $n$  puntos  $\zeta_i$ , distribuidos de manera no uniforme en el intervalo  $Z_1 \leq \zeta_i \leq Z_2$ , los cuales están dados por

$$\zeta_i = \frac{Z_2 - Z_1}{2} \kappa_i + \frac{Z_2 + Z_1}{2} . \quad (23)$$

Estos puntos de evaluación se escogen de tal forma que, la suma de los valores ponderados en la ecuación (21) entregue el valor exacto de la integral, cuando  $f(\zeta)$  puede ser representado por un polinomio de grado menor o igual a  $2n-1$  (Atkinson, 1989). Valores para las abscisas y factores de ponderación del método Gauss-Legendre, con hasta 12 cifras significativas, se pueden encontrar en Davis y Polonsky (1988). En la Tabla I, se muestran los valores de las abscisas y pesos utilizados en este método.

Una de las ventajas que proporciona el método de cuadratura de Gauss-Legendre es que  $\zeta_i$  está definido únicamente en intervalo abierto  $(-1,1)$  con lo cual se evita una singularidad esencial, la cual ocurre cuando el denominador de la ecuación (20) es idénticamente nulo. Ello ocurre, por ejemplo, cuando la posición del observador coincide con

Tabla I . Valores de las abscisas y pesos utilizados en el método de cuadratura de Gauss-Legendre

Abcisas $\bar{x} z_j$	Pesos $w_j$
0.148874338981631	0.295524224714753
0.433395394129247	0.269266719309996
0.679409568299024	0.219086362515982
0.865063366688985	0.149451349150581
0.973906528517172	0.066671344308688

la parte superior o inferior del prisma. Esta característica permite entonces la implementación de un algoritmo rápido y preciso.

Este método se ilustra con un ejemplo en el cual se ha utilizado un prisma bidimensional que se extiende *ad infinite* a lo largo del eje  $y$ , las unidades de longitud en los ejes  $x$  y  $z$  se dan en km y la densidad en  $\text{g/cm}^3$ . Sobre el eje  $x$ , el prisma se extiende desde  $X_1 = -3$  hasta  $X_2 = 3$  y sobre el eje  $z$  entre  $Z_1 = -1$  y  $Z_2 = 1$ , la densidad como función de la profundidad se describe mediante un polinomio de cuarto grado dado por

$$\sigma(z) = 1 - 4z + 6z^2 - 4z^3 + z^4 \quad (24)$$

La Figura 5 muestra una gráfica de la densidad descrita por la ecuación (24). El efecto gravimétrico  $g$  se calculó sobre una malla regular, o matriz, de 21 columnas por 21 renglones, con nodos igualmente espaciados a cada km sobre el plano  $x$ - $z$  en  $y = 0$ ; el resultado se presenta en la Figura 6 como un mapa de contornos de igual valor de gravedad.

La particular selección de  $\sigma(z)$  nos permite verificar algunas de las características esperadas en la anomalía gravimétrica. Nótese por ejemplo la simetría par de la anomalía con respecto al centro de masa del prisma, localizado en el origen de coordenadas del sistema. La anomalía es idénticamente nula ( $g = 0$ ) cuando coincide con el eje de simetría de masa del prisma ( $z = 0$ ) y que, con respecto a dicho eje ( $z = 0$ ) la anomalía tiene una simetría impar; esto es: cuando la posición del observador es tal que  $z_0 < 0$ , el signo de  $g$  concuerda con el de la densidad del prisma; por el contrario, cuando la posición del observador es tal que  $z_0 > 0$ ,  $g$  es negativa.

La densidad en el subsuelo puede ser definida de acuerdo con datos existentes, p. ej. registros de pozo o mediante extrapolación de la densidad observada en estructuras

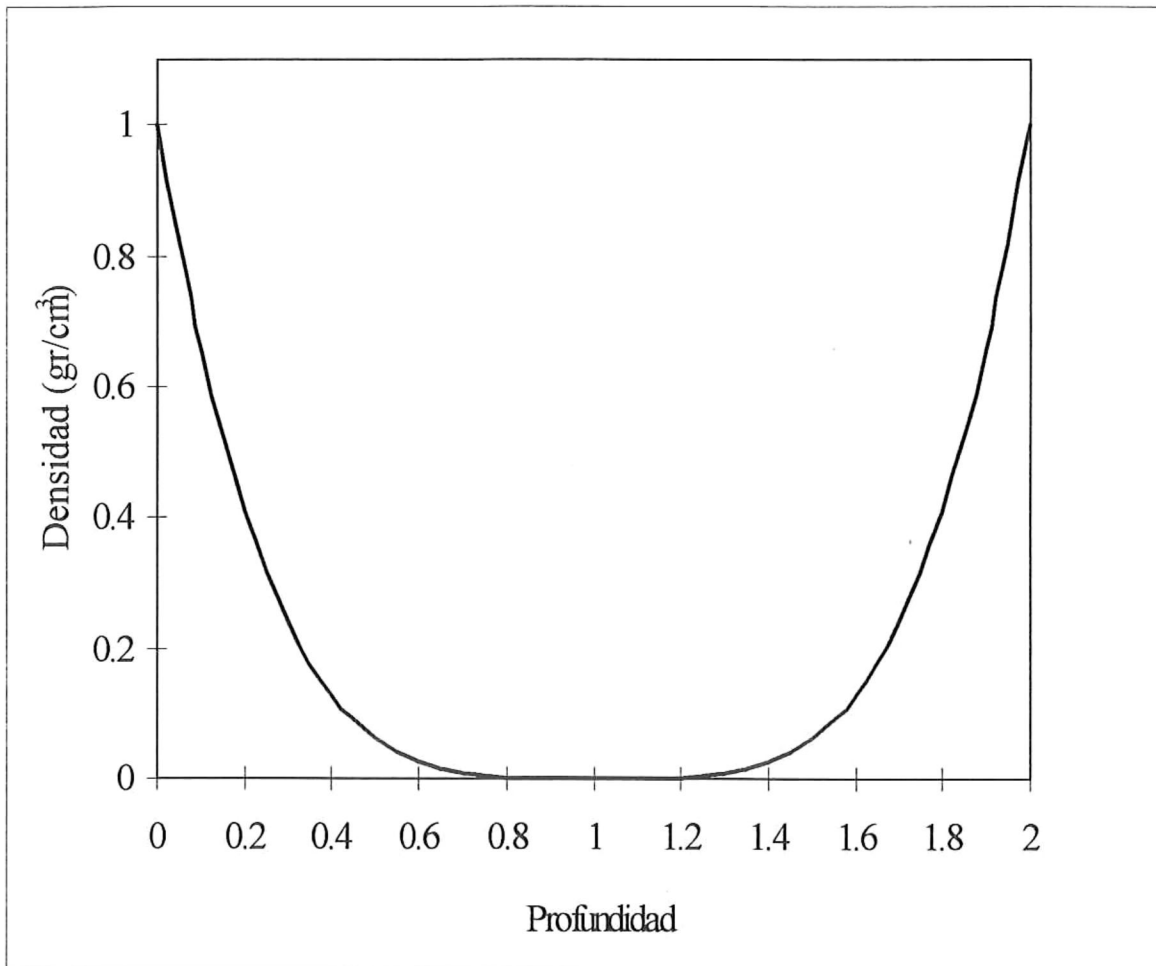


Figura 5. Gráfica de la densidad, como función de la profundidad.

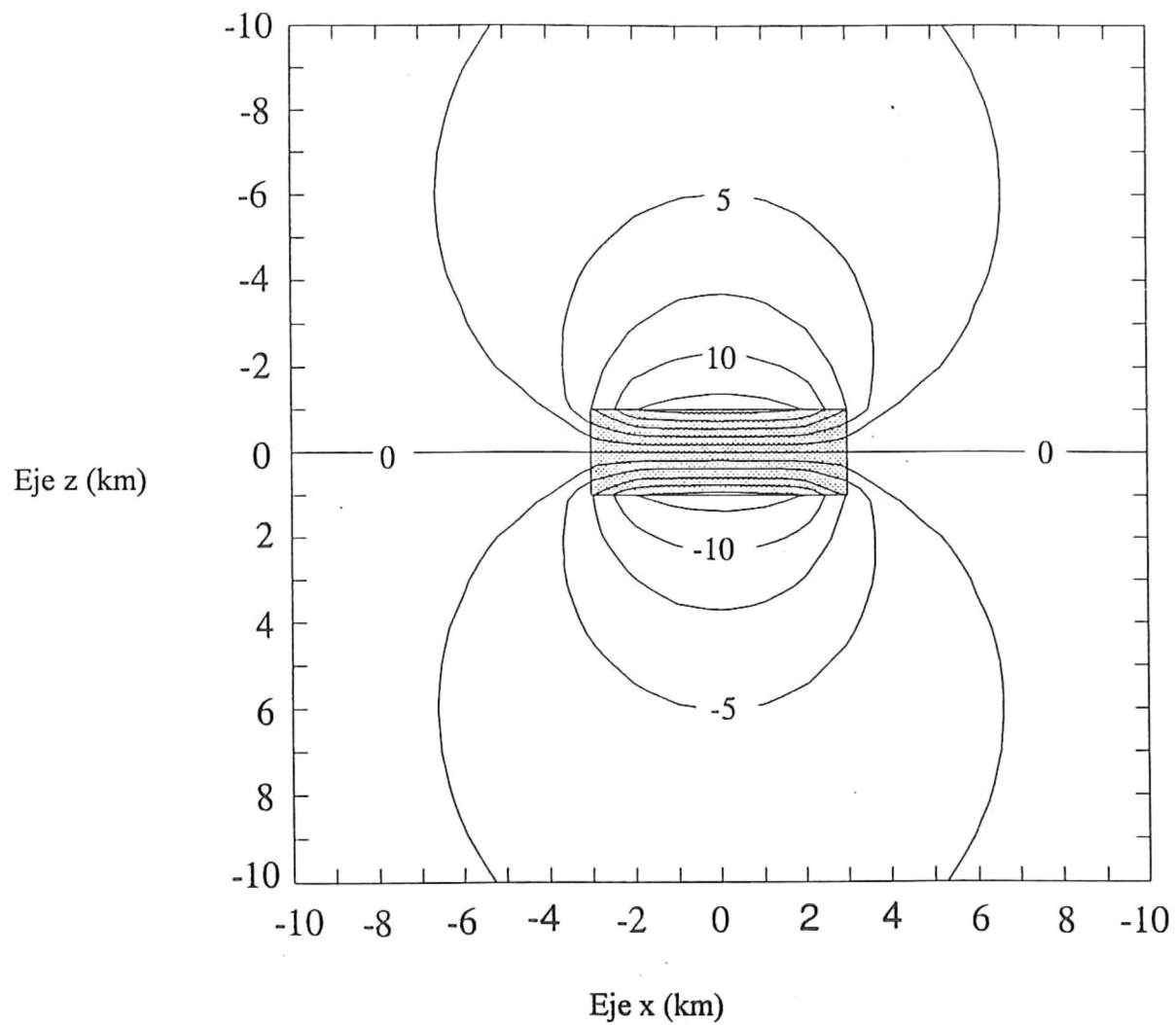


Figura 6. Mapa de contornos de los valores de gravedad, sobre el plano  $x-z$  en  $y = 0$ .

geológicas que afloran en superficie cerca del área de estudio. De esta manera podemos crear una función empírica de la densidad como función de la profundidad, ajustando los registros de densidad con un polinomio de grado adecuado.

## **V. METODOLOGÍA PARA LA SIMULACIÓN**

En este capítulo se describen las herramientas de software, el equipo de cómputo utilizado, y la metodología utilizada en el análisis y diseño del software.

### **V.1. Infraestructura de software y equipo de cómputo**

El análisis del sistema se realizó tomando en cuenta las características del equipo de cómputo y herramientas de software, disponibles en CICESE(1993), así como los requerimientos del sistema para el desarrollo de la simulación. A continuación se describen las herramientas utilizadas.

Para la implementación se contó con las estaciones de trabajo de la red CICESE disponibles en la División de Ciencias de la Tierra. Estas estaciones de trabajo son minicomputadoras Sun o Sparc con sistema operativo Unix 4.1.1. La capacidad de memoria en disco disponible para el usuario promedio es de 30 Mbytes. Dependiendo de la estación de trabajo, se cuenta con una capacidad de memoria RAM de hasta 128 Mbytes; la resolución gráfica de los monitores de 1152 X 900 pixeles. Las interfaces Openlook , X11 y SunView, proporcionan acceso amigable al sistema para codificar la aplicación y ejecutarla. Además se tienen unidades de cartucho de 60 y 150 MBytes, dispositivos para lectura en disco compacto (CD); una impresora térmica a color QMS Postscript, y una impresora Postscript Láser Sparc.

Se utilizó el lenguaje de programación C, este es un lenguaje de propósito general (Kernighan, 1978), que cuenta con características de transportabilidad y modificabilidad (Feibel, 1990).

Por otra parte X Window es una herramienta de software estándar utilizada para desarrollar interfaces gráficas de usuario sobre estaciones de trabajo y computadoras personales bajo Unix y otros sistemas operativos. El sistema X Window se ha convertido en un sistema de ventanas estándar independiente del hardware y del sistema operativo (Reiss, 1993). Se trata de una utilidad desarrollada para sistemas gráficos de alta resolución tanto en color como en monocromo.

Debido a la independencia de dispositivos (equipo de cómputo), las interfaces gráficas de usuario desarrolladas, pueden ser ejecutadas sin problema en plataformas de hardware distintas.

## **V.2. Metodología de Jackson**

Para la simulación del sistema propuesto es fundamental analizar el sistema y diseñar una solución, por medio de una serie de pasos, facilitando la implementación en un lenguaje de programación.

Para el análisis y el diseño de software, Jackson ha desarrollado una metodología orientada a la estructura de los datos tomados de los objetos del mundo real y sus manifestaciones. La metodología de Jackson requiere de una serie de etapas, las cuales se describen a continuación:

i) Etapa 1: se realiza un análisis general del sistema. Con base en dicho análisis se definen las entidades del sistema y las acciones que las afectan.

ii) Etapa 2: las entidades se representan, siguiendo un orden jerárquico, mediante un diagrama de entidades, en el cual, las acciones son aplicadas a una entidad en forma de una secuencia, una selección, o una iteración.

iii) Etapa 3: se plantea un modelo inicial en donde se muestra el flujo de datos de entrada, su interacción con el sistema y la respuesta del sistema.

iv) Etapa 4: se caracteriza la interacción entre las entidades y las acciones mediante funciones, las cuales representan una serie de eventos que necesitan ocurrir para ejecutar una acción determinada sobre las entidades.

v) Etapa 5: se analiza la sincronía de las acciones, lo cual permite detectar inconsistencias entre los eventos.

vi) Etapa 6: se requiere elaborar un pseudocódigo, en el cual, mediante un lenguaje sencillo y estructurado, se escriben los pasos para llevar a cabo las acciones, marcando la secuencia, los ciclos y las decisiones.

## VI. RESULTADOS

### VI.1 ANÁLISIS DEL SISTEMA GEOFÍSICO

En este capítulo se aplica la metodología de Jackson en el análisis del sistema geofísico.

#### VI.1.1. Análisis de entidades y acciones que afectan las entidades

La masa anómala y su efecto gravimétrico, se representarán gráficamente en un ambiente de ventanas. En la ventana gráfica se toma en cuenta el transecto por donde pasa un observador, el transecto está limitado entre los puntos  $x_1$  y  $x_2$ , entre los cuales se realizan mediciones del efecto gravimétrico.

Las entidades más importantes para el sistema son las masas anómalas y la gráfica de la función que representa el efecto gravimétrico de éstas. Las masa anómalas pueden ser una esfera ó un prisma rectangular. Otras entidades, además, son las variables de la esfera (el radio, profundidad a su centro de masa y su densidad), y las variables del prisma (el espesor y los coeficientes de la función que describen la densidad).

Las acciones que tendrán un efecto directo sobre las entidades son los cambios en los valores del radio, de la profundidad y de la densidad en el caso de la esfera, y en el caso del prisma afectarán los cambios de los valores de las constantes para el cálculo de la densidad.

#### VI.1.2. Diagrama de estructuras de entidades

En el diagrama mostrado en la Figura 7, se describen las acciones que afectan en la entidad o a causa de una entidad, siguiendo un orden específico. Esta representación del

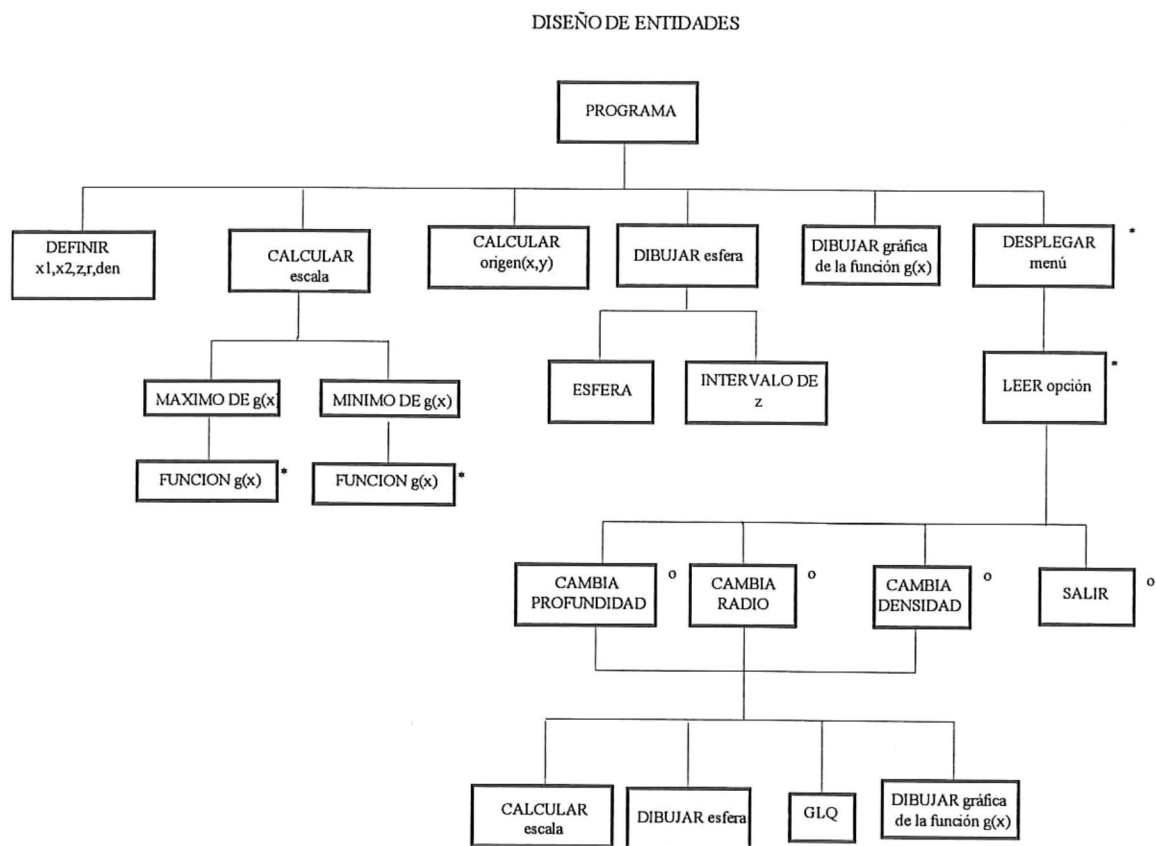


Figura 7. Diseño de estructuras de entidades.

mundo real es más precisa que una simple lista de acciones y entidades. Se crea un diagrama de estructura para cada entidad y otro (diagrama) para todas las entidades.

### VI.1.3. Modelo inicial del análisis

En la Figura 8 se muestra el modelo inicial del análisis del sistema, en el cual se representa la corriente o flujo de datos. En esta abstracción, los datos de entrada representan eventos que ocurren en el mundo real; el programa de simulación a su vez representa un sistema que interactuando con los eventos del mundo real nos presenta un nuevo escenario representado en los datos de salida.

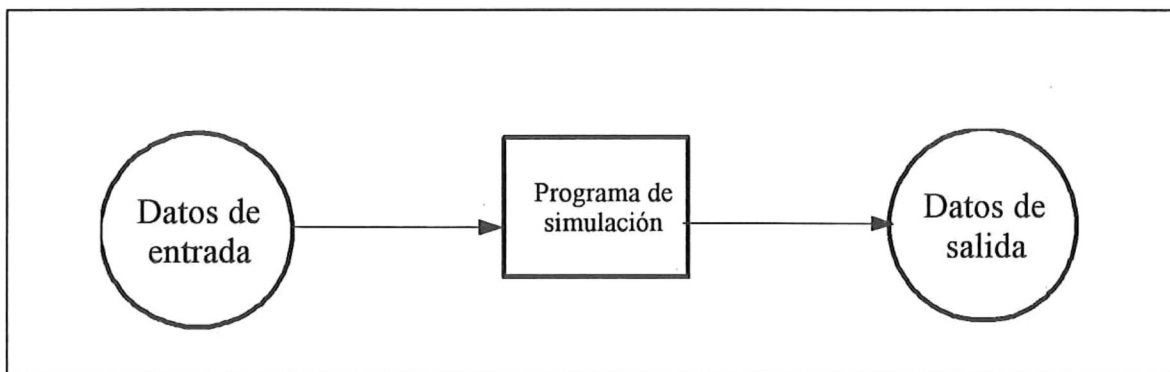


Figura 8. Modelo inicial del análisis del sistema.

## **VI.2. DISEÑO DEL SISTEMA**

### **VI.2.1. Funciones**

Para obtener los datos de salida a partir de los datos de entrada, tomando en cuenta las acciones mencionadas en el análisis de la metodología, es indispensable que un conjunto de funciones sean ejecutadas para llevar a cabo cada acción.

Algunas funciones sirven para definir valores de las entidades por medio de una serie de operaciones, otras nos despliegan la salida de datos, y otras detectan algún evento ocurrido en los dispositivos de hardware. Las funciones se presentan en el diccionario de datos, ya que así se mostrará el nombre y tipo de la función, las variables que interactúan en ella y un texto narrativo de lo que realiza. De esta forma se cuenta con mas información para lograr una mejor implementación.

### **VI.2.2. Sincronización**

Hay eventos que se espera que sucedan dentro del sistema, como por ejemplo: pulsación de un botón o finalización de un ciclo para que ocurra alguna de las acciones del sistema. De esta manera, a base de eventos se marca la sincronización dentro del sistema. Cada evento que ocurre tiene una acción relacionada con la entrada y salida de los datos.

### **VI.2.3. Implementación**

En los siguientes pseudocódigos, para las anomalías de la esfera y el prisma, respectivamente, se indican esta serie de pasos.

### VI.2.3.1. Pseudocódigo para la anomalía de la esfera

Módulo principal

Inicio principal

1.- Definir valores del transecto del observador ( $x_1, x_2$ ).

2.- Definir valor de profundidad para la esfera

3.- Definir valor de radio

4.- Definir valor de densidad

5.- Calcular el origen

6.- Calcular escala

7.- Dibujar esfera

8.- Dibujar gráfica de efecto gravimétrico

9.- Desplegar menú de opciones

10.- Leer opción

11.- Repite mientras opción sea diferente de salir

A: Si opción es cambiar radio

entonces leer valor de radio

B: Fsi

C: Si opción es cambiar profundidad

entonces leer valor de profundidad

D: Fsi

E: Si opción es cambiar densidad

entonces leer valor de densidad

F: Fsi

G: Si opción es salir

entonces salir del sistema

H: Fsi

I: Calcular escala

J: Dibujar esfera

K: Dibujar gráfica de efecto gravimétrico

L: Desplegar menú de opciones

M: Leer opción

12.- Fin repite

13.- Fin principal

### VI.2.3.2. Pseudocódigo para la anomalía del prisma

Módulo principal

Inicio principal

- 1.- Definir valores del transecto del observador ( $x_1=-50$ ,  $x_2=50$ ).
- 2.- Definir valor de profundidad para el prisma
- 3.- Definir valor de constantes para el cálculo de densidad
- 4.- Calcular origen0 y origen1
- 5.- Calcular valores de la función del efecto gravimétrico
- 6.- Calcular escala
- 7.- Dibujar prisma
- 8.- Dibujar gráfica de efecto gravimétrico
- 9.- Desplegar menú de opciones
- 10.- Leer opción
- 11.- Repite mientras opción sea diferente de salir

A: Si opción es cambiar constantes

entonces

desplegar menú de polinomios

elegir tipo de polinomio

B: Si opción es polinomio de primer orden

entonces leer  $c_0$

C: Fsi

D: Si opción es polinomio de segundo orden

entonces leer  $c_0, c_1$

E: Fsi

F: Si opción es polinomio de tercer orden

entonces leer  $c_0, c_1, c_2$

G: Fsi

H: Si opción es polinomio de cuarto orden

entonces leer  $c_0, c_1, c_2, c_3, c_4$

I: Fsi

J: Si opción es polinomio de orden exponencial

entonces leer  $c_0, c_1$

K: Fsi

L: Fsi

M: Si opción es Seleccionar polinomio

entonces elegir tipo de polinomio

N: Fsi

O: Si opción es salir

entonces salir del sistema

P: Fsi

Q: Hacer matriz de valores ( $x, grav$ )

R: Calcular escala

S: Dibujar prisma

T: Dibujar gráfica de efecto gravimétrico

U: Desplegar menú de opciones

V: Leer opción

12.- Fin repite

13.- Fin de módulo principal

#### **VI.2.4. Interfaz de Entrada y Salida de Datos**

Para el desarrollo de la interfaz era necesario familiarizarse con X Window. La mayoría de estatutos y características del X Window, que fueron de utilidad en el desarrollo de la simulación, fueron tomados de Nye(1992).

Los datos de entrada del usuario se capturan, desde el monitor, con el uso de menús, teclado y del ratón. Para la modificación del radio y profundidad de la esfera, dentro de la ventana de ésta, se utiliza la pulsación del ratón. Dependiendo de la opción seleccionada se ejecutan una serie de funciones que afectan las características de la masa anómala y la gráfica de la función del efecto gravimétrico respectivamente.

La salida es la gráfica de la función del efecto de gravedad y la figura de la masa anómala. El despliegue de la salida de datos se producirá en un conjunto de ventanas, con diferentes características en tipos de entradas, tamaño, color y ambiente gráfico. El ambiente de ventanas estará conformado por una ventana para la gráfica del efecto

gravimétrico, otra más para la masa anómala y los menús también estarán compuestos de ventanas para el despliegue de las opciones; todas ellas dentro de una ventana raíz.

En la Figura 9 se muestra una descripción de la interfaz de salida del sistema y una de las ventana nos sirve para mostrar el comportamiento del modelo y otra de las ventanas se utiliza para desplegar la esfera respectivamente al modelo usado. La ventana principal contiene a las dos ventanas mencionadas anteriormente y el menú de opciones.

En el menú se tienen las siguientes opciones: cambiar el valor de profundidad, cambiar el valor de radio, cambiar el valor de densidad y salir del programa.

Las opciones cambiar z, cambiar r y cambiar densidad se realizan por medio de procedimientos antes mencionados. La opción salir se realiza con una instrucción del propio lenguaje.

#### **VI.2.5. Diccionario de Datos**

El diccionario de datos se presenta en forma tabular, en donde se indican el nombre del elemento, su descripción, tipo de elemento y su rango o valor. En la lista de procesos se indica, tanto para la esfera como para el prisma, la descripción del proceso, los datos de entrada y los datos de salida.

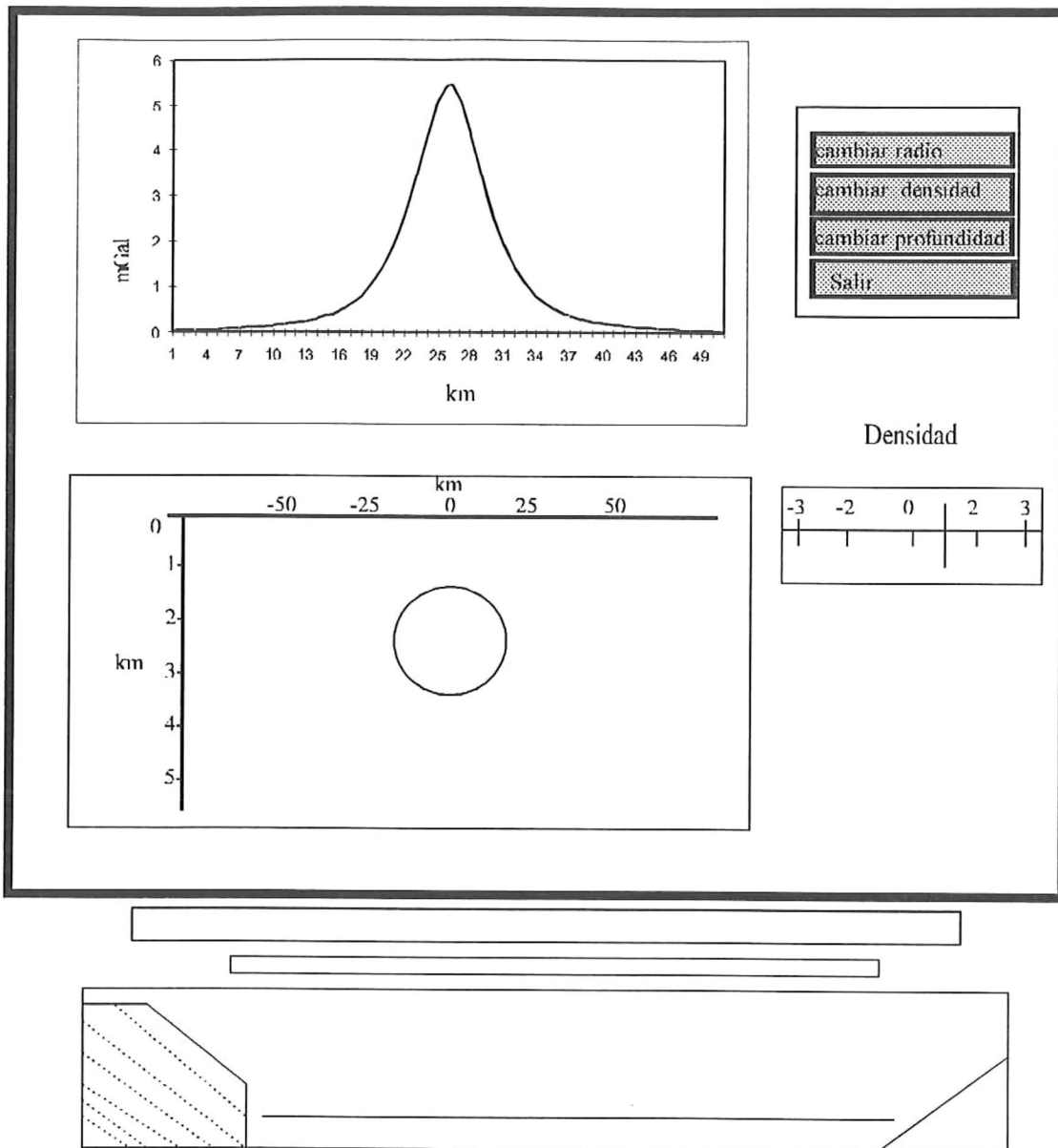


Figura 9. Diseño de interfaz gráfica .

## VI.2.5.1. Lista de elementos

ELEMENTO	DESCRIPCIÓN	TIPO	RANGO
Radio	Número real que indica el valor del radio de la esfera.	Flotante	(0,50)
x0, x1	Los valores x0 y x1, representan los números reales del inicio y el final del transecto del observador respectivamente.	Arreglo de 2 números flotantes	x0=-50 x1=50
z	Es el valor de la profundidad desde el nivel de la superficie hasta el centro de la esfera.	Flotante	(0.5,100)
den	Es el valor de la densidad de la esfera.	Flotante	(-3,3)
esc	Es el valor de la escala que se usa en el cálculo del mapeo de la gráfica y la esfera.	Flotante	
esfera	Es una circunferencia representativa de la masa anómala.	Arco	
grav	Es el valor del efecto gravimétrico para un punto de x0 a x1	Flotante	
gráfica g(x)	Es el conjunto de (x(i),grav(i)) unidos por líneas que representa la función del efecto gravimétrico en escala esc.	Conjunto de líneas.	
menú	Es un arreglo que contiene las 3 opciones de los cambios en los parámetros de la esfera y la	Arreglo de 4 cadenas	

	opción de salida.		
--	-------------------	--	--

### VI.2.5.2. Lista de Procesos para la esfera

Proceso: Cambiar radio
Descripción: Obtener un valor nuevo para el radio de la esfera
Entrada: Valores de (x, y) donde el ratón es colocado
Salida: Radio, esfera y efecto gravimétrico

Proceso: Cambiar profundidad
Descripción: Leer un nuevo valor para la profundidad de la esfera
Entrada: Valores de(x, y) de la colocación del ratón, en el monitor.
Salida: Profundidad, esfera y gráfica de efecto gravimétrico

Proceso: Cambiar densidad
Descripción: Leer nuevo valor para la densidad de la esfera.
Entrada: Valores (x, y) de la lectura del ratón
Salida: Densidad, efecto gravimétrico

Proceso: Calcular origen para gráfica
Descripción: Obtener valores de (x, y) para el punto de inicio de la gráfica, en la ventana

correspondiente de acuerdo a los valores de longitud y ancho de dicha ventana
Entrada: ancho_ventana, largo_ventana, escala
Salida: origenx, origeny

Proceso: Calcular escala
Descripción: Obtener escala para el despliegue de la gráfica del efecto gravimétrico
Entrada: ancho_ventana, largo_ventana, maximox , minimox, maximogx, minimogx
Salida: escala

Proceso: Dibujar gráfica de efecto gravimétrico
Descripción: Unión de líneas con nodos en los puntos $(x(i),g(x(i)))$
Entrada: origenx, origeny, maximox , minimox, maximogx, minimogx, escala, fijo
Salida: Gráfica del efecto gravimétrico

El programa computacional que lleva a cabo la simulación, permite mostrar en una ventana gráfica el comportamiento del sistema geofísico, cuando ocurren cambios en sus parámetros. En la Figura 10 y 11 se muestran las ventanas, en donde aparecen la figura geométrica de la masa anómala y su correspondiente efecto gravimétrico. Adicionalmente, la ventana incluye un menú de opciones que permiten modificar los parámetros del modelo, viéndose en pantalla el modelo con el cambio realizado. Este es

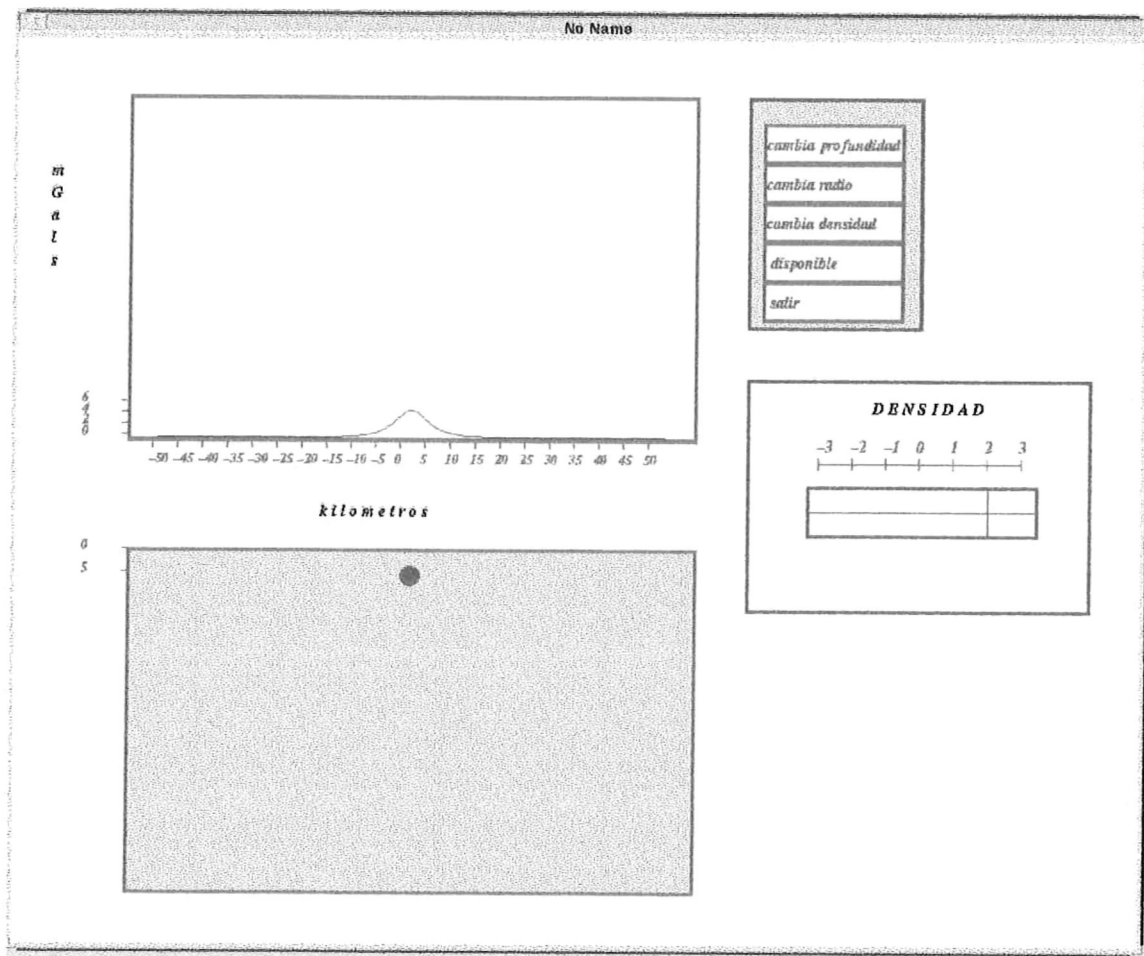


Figura 10. Interfaz gráfica del sistema para la esfera.

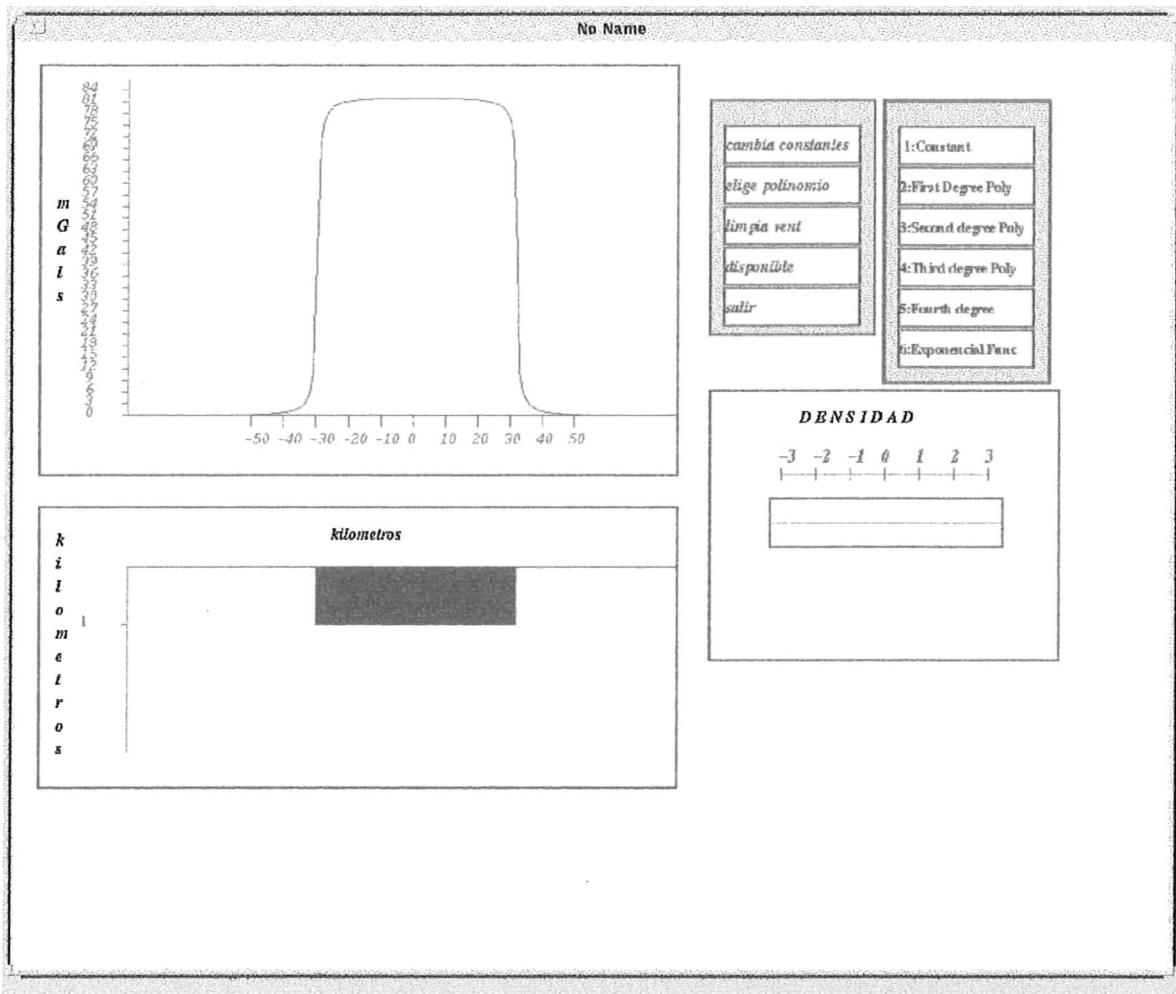


Figura 11. Interfaz gráfica del sistema para el prisma.

el sistema básico del que se parte para realizar otros modelos (Figura 11). El programa se compone de módulos que realizan procedimientos independientes entre sí , por lo que al agregar los nuevos modelos no afecta a los que se tienen.

## VII. DISCUSIÓN

Por medio de la simulación, se puede observar el efecto de cambios internos y externos del sistema geofísico, y al hacer alteraciones en el modelo del sistema, nos llevará a un mejor entendimiento de los efectos que éstas provocan en el comportamiento del sistema.

La simulación puede ser una herramienta para experimentar con nuevas situaciones, sobre las cuales se tiene poca o ninguna información de lo que podría ocurrir en el sistema geofísico de anomalías gravimétricas. A través de esta experimentación se pueden anticipar a posibles resultados no previstos. De esta forma el usuario obtiene experiencia en el diagnóstico del modelo más apropiado para interpretar la anomalía observada.

Para la simulación del sistema propuesto es fundamental analizar el sistema y diseñar una solución, siguiendo una serie de pasos, que faciliten la implementación en un lenguaje de programación. Existen diversas metodologías para el análisis y diseño de software, que nos dan una guía de los pasos a seguir para describir la representación del sistema, las salidas y entradas de información, archivos y procedimientos.

En cuanto a metodologías hay distintas preferencias para llevar a cabo el diseño. Algunas metodologías están orientadas al manejo de información o a la representación de objetos (Pressman, 1992). Warnier (1981, en Pressman, 1992) propone analizar y diseñar con un enfoque orientado al resultado, o salida, del sistema. La metodología de Jackson

(1983, en Pressman, 1992) en particular, se enfoca al manejo de estructuras de los datos. Dependiendo del tipo de sistema se elige la metodología que se apegue mejor a la información con que se cuenta (entrada de datos) y al resultado esperado del sistema (salida de datos). Tomando en cuenta las características del sistema geofísico se utilizó la metodología de Jackson para el análisis y diseño del sistema.

Algunos programas computacionales, que sirven de apoyo a investigadores, representan los resultados en forma de gráficas, o bien, en archivos de datos numéricos que se trasladan a paqueterías para visualizar los resultados, por lo que se invierte más tiempo de trabajo en la interpretación de los datos obtenidos.

Las características de un lenguaje son importantes en el momento de elegir alguno para la implementación del sistema. Una de estas características es la transportabilidad. La transportabilidad nos permite la ejecución de programas en otras plataformas de hardware, lo cual puede ser de utilidad si hubiera cambios de equipo de cómputo, o para su implementación fuera del CICESE, en este caso. La modificabilidad permite hacer cambios al código, de una manera sencilla. Otra ventaja es la facilidad de realizar los programas en módulos, lo cual facilita la legibilidad y desarrollo del código, reduciendo sustancialmente el tiempo de verificación, así como la creación de módulos que podrían ser de utilidad.

## VIII. CONCLUSIONES

En este trabajo se ha desarrollado una infraestructura computacional básica que permite simular dos modelos geofísicos, mediante el uso de áreas o ventanas gráficas. En este ambiente se utilizan menús que permiten modificar fácilmente, los parámetros de ambos modelos geofísicos con ayuda del ratón.

La infraestructura computacional desarrollada nos permite visualizar la masa anómala y su efecto gravimétrico, siendo posible efectuar cambios en los parámetros de los modelos y visualizarlos en forma inmediata. Los cambios en los parámetros se realizan mediante menús, a los cuales se accede mediante el ratón.

En el modelo de la esfera, los parámetros posibles de modificar son: el radio, la profundidad al centro de la esfera y su densidad. Para el cambio en el radio, la densidad y la profundidad, se cuentan con funciones que leen coordenadas en pantalla de la posición  $(x, y)$  del ratón y de esta manera hacen el mapeo de los datos, para así obtener el nuevo valor del radio, de la densidad o de la profundidad al centro de la esfera.

En el modelo del prisma se cuenta con funciones de polinomios para cambiar el valor de la densidad en el mismo. Los cambios se producen esencialmente en los coeficientes que corresponden a los distintos tipos de funciones, que caracterizan a la densidad.

La infraestructura computacional desarrollada nos permite realizar cambios en los modelos geofísicos, en un ambiente de ventanas, sin tener que abandonar la plataforma para pasar a otro paquete computacional con el fin de visualizar los resultados.

La observación del modelo geofísico y su respectiva función gravimétrica facilita el trabajo de los investigadores o estudiantes en el área de geofísica para interpretar propiedades físicas del subsuelo.

El ambiente de la simulación, puede emplearse como plataforma para visualizar el comportamiento de otros modelos geofísicos, con el sistema ya existente.

## LITERATURA CITADA

- Athy, L.F. 1930. Density, porosity and compaction of sedimentary rocks: Bull. Am. Assoc. Pet. Geol. v. 14, No.11-14 pp.
- Atkinson, K. E. 1989. An introduction to numerical analysis: John Wiley & Sons, Inc. New York 2nd ed. 693 pp.
- Chai, Y., and Hinze W. J. 1988. Gravity inversion of an interface above which the density contrast varies exponentially with depth: Geophysics, v. 53, No. 6. 837-845 pp.
- Cordell, L. 1973, Gravity analysis using an exponential density-depth function- San Jacinto graben, California: Geophysics, v. 38, No. 4. 684-690 pp.
- Coss R. 1982. Simulación, un enfoque práctico. Ed. Limusa. 1a. edición. México. 7 pp.
- Davis, P. J., and Polonsky, I. 1988. Numerical interpolation, differentiation and integration, *in* Abramowitz, M., and Stegun, I. A., eds., Handbook of Mathematical functions with formulas, graphs, and mathematical tables: Dover Publ. Inc. 875-924 pp.
- Feibel W. 1990. Using Ansi C in Unix. McGraw-Hill. Berkeley, Ca. U.S.A. 8,9 pp.
- García-Abdeslem, J. 1992. Gravitational attraction of a rectangular prism with depth-dependent density: Geophysics, v. 57. No. 3. 470-473 pp.
- García-Abdeslem, J. 1996 . gl2d: A FORTRAN program to compute the gravity anomaly of a 2-D prism where density varies as a function of depth: Computers & Geosciences, 22. No. 7. 823-826 pp.

- Kernighan B.W. y D. Ritchie. 1978. The C programming language. Prentice-Hall. New Jersey, U.S.A. 1 pp.
- Law, A.M., y W. Kelton. 1991. Simulation modeling and analysis. 2nd. Ed. McGraw- Hill. U.S.A.
- Murthy, I. V. R., and Rao, D. B. 1979. Gravity anomalies of two-dimensional bodies of irregular cross-section with density contrast varying with depth: Geophysics. v. 44. No. 9. 1525-1530 pp.
- Nye A. 1992. Xlib Programming Manual. 3rd. Edition. O'Reilly & Associates, Inc. U.S.A.
- Pressman, R. S. 1992. Software engineering: a practitioner's approach. 3rd Ed. McGraw-Hill. U.S.A. 267-312, 429-456 pp.
- Rao, D. B. 1990. Analysis of gravity anomalies of sedimentary basins by an asymmetrical trapezoidal model with quadratic density function: Geophysics. v. 55, No. 2. 226-231 pp.
- Reiss L. y J. Radin. 1993. Aplique X Window. McGraw-Hill. México.
- Shannon, R. E. 1988. Simulación de sistemas : diseño, desarrollo e implementación . 3era. Ed. Trillas. México. 11,165 pp.
- Telford W., Geldar L., Sheriff R.& Keys D. 1976. Applied Geophysics. 1st. edition. Cambridge University Press. Cambridge. Great Britain. 7 pp.
- Tremblay J. y R. Bunt. 1988. Introducción a la ciencia de las computadoras enfoque algorítmico. McGrawHill. 1a. Edición. México, D.F. 424 pp.

## APÉNDICE A

### PROGRAMAS FUENTES Y BIBLIOTECAS

A continuación se presenta el código en lenguaje C con una interfaz gráfica en el sistemaX Window. El archivo prisma.c requiere de las librerías glq.h y prisma.h. El archivo esfera.c requiere de la librería esfera.h.

*Inicio de prisma.h*

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

int i;
static char *opciones[5]={"cambia constantes","elige polinomio","limpia vent","disponible","salir"};
char *cero[2]={"0",""};
char *caden[6];

void Ejes(d,w,gc,x1,y1,x2,y2) /*Se dibujan las líneas que sirven como ejes tanto en el despliegue de la gráfica como del prisma*/
Display *d;
Window w;
GC gc;
int x1,x2,y1,y2;
{
  XDrawLine(d,w,gc,x1,y1,x2,y1);
  XDrawLine(d,w,gc,x1,y1,x1,y2);
}

float menor(x1,x2) /*Regresa el valor menor de dos valores*/
float x1,x2;
{float vmenor;
  if (x1<x2) vmenor=x1;
  else vmenor=x2;
  return(vmenor);
}

float mayor(x1,x2) /*Regresa el valor mayor de dos valores*/
float x1,x2;
{float vmayor;
  if (x1>x2)
    vmayor=x1;
  else vmayor=x2;
  return(vmayor);
}

float escalar(x1,x2, fx1, fx2, ancho, largo) /* Para calcular la escala aplicada en la gráfica de la función*/
float x1,x2,fx1,fx2;
int ancho,largo;
float x1,x2,fx1,fx2;
{
  float distanciax,distanciafx,escx,escfx,escala;
```

```

distanciax=x2-x1;
distanciax+=distanciax*0.1;
distanciafx= (float) fx2-fx1;
distanciafx+=distanciafx*0.1;
escx=ancho/distanciax;
escfx=largo/distanciafx;
escala=menor(escx,escfx);
return(escala);
}

void GrafFuncion(d,w,gc,xmin,xmax,fxmin,fxmax,esc,coordx,coordy,ejem,xfij) /*Se dibuja la gráfica correspondiente a la */
Display *d; /*función gravimétrica*/
Window w;
GC gc;
float xmin,xmax,fxmin,fxmax,esc,ejem[101][2];
int coordx,coordy,xfij;
{
float xi,fxi,tempfx=0,tempx=0;
int intx1,intfx1,intx2,intfx2;
    if (fxmin<0) tempfx=-1*(fxmin*esc);
    else tempfx=-1*(fxmin*esc);
    if (xmin<0) tempx=-1*(xmin*esc);
    else tempx=-1*(xmin*esc);\

intx1=(int)(coordx+tempx+esc*ejem[0][0]);
intx1+=xfij;
intfx1=(int)(coordy-(tempfx+esc*ejem[0][1]));
for(i=1;i<=100;i++)

{ xi=ejem[i][0];
fxi=ejem[i][1];
intx2=(int)(coordx +tempx+esc*xi);
intx2+=xfij;
intfx2=(int)(coordy -(tempfx+ esc*fxi));
XDrawLine(d,w,gc,intx1,intfx1,intx2,intfx2);
intx1=intx2;
intfx1=intfx2;
}
}

void PonOpcion(d,w,gc1,gc2) /*Se resalta la última opción elegida */
Display *d;
Window w;
GC gc1,gc2;
{
XFillRectangle(d,w,gc1,0,0,115,30);
XDrawString(d,w,gc2,0,20,opciones[i],strlen(opciones[i]));
}

itoa(n,s) /* Función que te convierte un valor entero a una cadena de caracteres */
char s[];
int n;
{
int i,signo;
if ((signo=n)<0) n=-n;
i=0;
do { s[i++]=n % 10 +'0';
} while ((n/=10)>0);
if (signo<0) s[i++]='-';
s[i]='\0';
reverse(s);
}

reverse(s) /*Se obtiene el reverso de una cadena */
char s[];
{ int c,i,j;

```

```

for (i=0,j=strlen(s)-1;i<j;i++j--)
    {
        c=s[i];
        s[i]=s[j];
        s[j]=c;
    }
}

void IntervaloY(d,w,gc,max,min,esc,origenx,origeny) /*Para desplegar el Intervalo en el eje "y" del efecto gravimétrico*/
Display *d;
Window w;
GC gc;
float max,min,esc;
int origenx,origeny;
{
    float distancia;
    int primera,segunda,contador,gx_entera,num_intervalos,num_graf;
    char *cad[1];
    cad[0]="3";
    contador=0;
    primera=origenx-5;
    segunda=origenx;
    distancia= esc*(max-min);
    num_intervalos=(int) (distancia/10);
    gx_entera=origeny;
    num_graf=(int) min;
    while (contador<=num_intervalos+1)
    {
        XDrawLine(d,w,gc,primera,gx_entera,segunda,gx_entera);
        itoa(num_graf,cad[0]);
        if ((strlen(cad[0]))==1)
            XDrawString(d,w,gc,segunda-36,gx_entera+2,cad[0],strlen(cad[0]));
        else
            XDrawString(d,w,gc,segunda-40,gx_entera+2,cad[0],strlen(cad[0]));
        num_graf+=(int) (10/esc);
        gx_entera-=10;
        contador++;
    }
}

void IntervaloX(d,w,gc,max,min,esc,origenx,origeny) /*Para desplegar el Intervalo en el eje "x" del efecto gravimétrico*/
Display *d;
Window w;
GC gc;
float max,min,esc;
int origenx,origeny;
{
    float distancia;
    int primera,segunda,contador,gx_entera,num_intervalos,num_graf,uno,dos,temp;
    char *cad[1];
    cad[0]="3";
    contador=0;
    primera=origeny;
    segunda=origeny+10;
    distancia= esc*(max-min);
    temp=(int)((500.0-distancia)/2);
    origenx+=temp;
    gx_entera=origenx;
    if (distancia>400) num_intervalos=20;
    else if (distancia < 200&& distancia>=100) num_intervalos=4;
        else if (distancia <100) num_intervalos=2;
            else num_intervalos=10;
    uno=(int)((max-min)/num_intervalos);
    dos=(int)((500-2*temp)/num_intervalos);
    num_graf=(int)min;
    while (contador<=num_intervalos)

```

```

    {
        XDrawLine(d,w,gc,gx_entera,primera,gx_entera,segunda);
        itoa(num_graf,cad[0]);
        XDrawString(d,w,gc,gx_entera-4,origeny+25,cad[0],strlen(cad[0]));
        num_graf+=(int)uno;
        gx_entera+=(int)dos;
        contador++;
    }
}

void IntervaloZ(d,w,gc,max,min,esc,origenx,origeny) /*Para desplegar el Intervalo de la profundidad (z) del prisma*/
Display *d;
Window w;
GC gc;
float max,min,esc;
int origenx,origeny;
{
    float distancia;
    int primera,segunda,contador,gx_entera,num_intervalos,num_graf;
    char *cad[1];
    cad[0]="3";
    primera=origenx-5;
    segunda=origenx;
    distancia=(int)(esc*(max-min));
    distancia=50.0;
    gx_entera=origeny+(int) distancia;
    num_graf=(int) max;
    itoa(num_graf,cad[0]);
    XDrawLine(d,w,gc,primera,gx_entera,segunda,gx_entera);
    XDrawString(d,w,gc,segunda-40,gx_entera+2,cad[0],strlen(cad[0]));
}

float dens(indice,zeta,c) /*Dependiendo del valor indice regresa el valor correspondiente al polinomio indicado*/
int indice;
float zeta,c[];
{
    float densi;
    switch(indice)
    {
        case 0: densi=c[0];
            break;
        case 1: densi=c[1]*zeta+c[0];
            break;
        case 2: densi=(c[2]*zeta+c[1])*zeta+c[0];
            break;
        case 3: densi=((c[3]*zeta+c[2])*zeta+c[1])*zeta+c[0];
            break;
        case 4: densi((((c[4]*zeta+c[3])*zeta+c[2])*zeta+c[1])*zeta+c[0]);
            break;
        case 5: densi=c[0]*exp(c[1]*zeta);
            break;
    }
    return(densi);
}

float CambiaDensidad(d,w,gc,w1,gc1,color) /* Para cambiar el valor de la densidad según el polinomio elegido*/
Display *d;
Window w,w1;
GC gc,gc1;
char *color;
{
    XEvent densidad;
    float den,q,temp;
    int u,x1,y1,y0,signo,nn,ni,cont,lent;
    char *punto=".",*neg="-";
    x1=0;

```

```

y0=0;
y1=40;
caden[0]="";
caden[1]="2";
caden[2]="";
caden[3]="0";
caden[4]="2";
caden[5]="0";
cont=0;
nn=0;
ni=0;
lent=signo=0;
den=0;
u=1;

while(u)
{
XNextEvent(d,&densidad);
if (densidad.xany.window==w)
switch(densidad.type)
{
case ButtonPress:
x1=densidad.xbutton.x;
den=x1*0.033333-3.0;
break;
}
XClearWindow(d,w);
XDrawLine(d,w,gc,x1,y0,x1,y1);
XDrawLine(d,w,gc,0,20,200,20);
if (densidad.xbutton.button==Button3) u=0; /*Si el botón elegido es el botón 3 termina de elefir el valor de densidad*/
else{
caden[2]="";
temp=den;
if (temp<0.0) { signo=1; temp*=-1;}
else { signo=0;}
nn=(int)temp%10;
itoa(nn,caden[1]);
q=temp-nn;
ni=(int)(q*10000);
itoa(ni,caden[2]);
lent=strlen(caden[2]);
cont=0;
XClearWindow(d,w1);
strcat(caden[1],punto);
if (lent<4)
{
cont=4-lent;
for (i=1;i<=cont;i++)
{strcat(cero[1],cero[0]);}
}
strcat(caden[1],caden[2]);
if (signo==1)
XDrawString(d,w1,gc1,5,20,neg,strlen(neg)); /* Se despliega el valor del coeficiente elegido */
XDrawString(d,w1,gc1,11,20,caden[1],strlen(caden[1]));
caden[1]="";nn=0;ni=0;cero[1]="";
}
}
return(den);
}
void PonDensidad(d,w,gc) /* Despliega en una ventana el valor de la densidad y el intervalo de (-3 , 3)*/
Display *d;
Window w;
GC gc;
{
char *cad[1];
int x0,y0,x1,y1,k;
cad[0]="3";

```

```

y0=90;
y1=130;
k=-3;
XSetLineAttributes(d,gc,1,LineSolid,CapButt,JoinBevel);
XDrawLine(d,w,gc,60,70,240,70);
for(i=60;i<=240;i+=30,k++)
  { XDrawLine(d,w,gc,i,65,i,75);
    itoa(k,cad[0]);
    XDrawString(d,w,gc,i-2,60,cad[0],strlen(cad[0]));
  }
}
void PonLinDen(d,w,gc) /* Es la línea eje de la densidad*/
Display *d;
Window w;
GC gc;
{ XDrawLine(d,w,gc,0,20,200,20); }

Fin de prisma.h

Inicio de glq.h

float glq(zi,xi,x0,z0,index1,c) /*Esta función regresa el valor del efecto gravimétrico para cada valor de x*/
float zi[],x0,z0,xi[],c[];
int index1;
{
float prof[10],w[10],zm,zr,sum,a,pot,s,zz,arg1,zeta,d,g;
int k,i;

prof[0]=-.973906528517172;
prof[1]=-.865063366688985;
prof[2]=-.679409568299024;
prof[3]=-.433395394129247;
prof[4]=-.148874338981631;
prof[5]=.148874338981631;
prof[6]=.433395394129247;
prof[7]=.679409568299024;
prof[8]=.865063366688985;
prof[9]=.973906528517172;

w[0]=.066671344308688;
w[1]=.149451349150581;
w[2]=.219086362515982;
w[3]=.269266719309996;
w[4]=.295524224714753;
w[5]=.295524224714753;
w[6]=.269266719309996;
w[7]=.219086362515982;
w[8]=.149451349150581;
w[9]=.066671344308688;

zm=0.5*(zi[1]+zi[0]);
zr=0.5*(zi[1]-zi[0]);
sum=0;

for (k=2;k>=1;k--)
  {
a=xi[k-1]-x0;
if (k==2) pot=(-1)*(-1); else pot=(-1);
s=0;

for (i=0;i<10;i++)
  {
zz=zr*prof[i]+zm;
arg1=a/(zz-z0);
zeta=zz-zi[0];
d=dens(index1,zeta,c);

```

```

        s=s+w[i]*atan(arg1)*d;
    }
    sum=sum+pot*s;
}
g=sum*6.6732*(zi[1]-zi[0]);
return(g);
}

```

*Fin de glq.h*

*Inicio de prisma.c*

```

#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <stdio.h>

#include "prisma.h"
#include "glq.h"
/* Declaración de variables*/
/*Declaración de variables que permiten el ambiente de ventanas del ambiente X Window*/
Display *d;
Window r,ventana[7],opcion[5],densidad[5],menu2,indice[6],constant[2];
GC grafic[7],grafopc[5][2],grafdensidad,grafmen,grafind[6][2],grafcons[2];
XEvent evento,even_ind;
XSetWindowAttributes att;
Font f1,f2,f3,f4;
/*Valores de cadenas para las opciones del menú de polinomios, laos coeficientes y etiquetas */
static char *tipocdes[7]={"", "1:Constant ", "2:First Degree Poly","3:Second degree Poly","4:Third degree Poly",
"5:Fourth degree","6:Exponencial Func" };

static char *constan[5]={ "C[1]=", "C[2]=", "C[3]=", "C[4]=", "C[5]="};
static char *etiql[17]={ "kilometros", "m", "G", "a", "l", "s",
"K", "i", "l", "o", "m", "e", "t", "r", "o", "s", "D E N S I D A D"};
unsigned long MyColor(); /* Los colores y la función que los inicializa para el ambiente del sistema*/
unsigned long beige,blue,red,gray,white,black;
FILE *out; /*archivo de salida de datos de los valores x y su respectivo valor de gravedad y la profundidad*/
static struct {
    int x,y,largo,ancho; /*Características de las ventanas : origen (x,y) y los valores de ancho y largo*/
} vent[7];
static struct {
    int x,y;
    } origen[3]; /*Valores que sirven como origen de despliegue y de intervalos*/
float a,escala,xi[2],zi[2],x0,z0,c[6],grav,ejem[101][2],q,t;
float minx,maxx,mingx,maxgx,tempx0,tempg;
int marco,m,n,h,j,l,k,indice1,poseje,bandera,flag,cons,nn,ni,signo,xfijo;
unsigned nd;

IniciaVentanas() /*Inicializa las ventanas del despliegue*/
{
    XClearWindow(d,ventana[0]);
    XClearWindow(d,ventana[1]);
    XClearWindow(d,ventana[2]);
    XClearWindow(d,ventana[3]);
    XSetForeground(d,grafic[5],blue);
    XSetForeground(d,grafic[6],blue);
    PonDensidad(d,ventana[5],grafic[5]);
    PonLinDen(d,ventana[6],grafic[6]);

    for(i=0;i<5;i++)
    {XSetForeground(d,grafopc[i][0],white);
    XSetForeground(d,grafopc[i][1],blue);
    PonOpcion(d,opcion[i],grafopc[i][0],grafopc[i][1]);
    }
}

menu_2() /*Se visualizan las opciones de polinomios*/

```

```

{
for (i=0;i<6;i++) {
    XSetForeground(d,grafind[i][0],white);
    XSetForeground(d,grafind[i][1],blue);
    XDrawString(d,indice[i],grafind[i][1],0,20,tipoctes[i+1],
    strlen(tipoctes[i+1]));
}
}

ven_cons() /*Para visualizar el valor de los coeficientes de un polinomio*/
{
XSetForeground(d,grafcons[0],blue);
XSetForeground(d,grafcons[1],blue);
XMapSubwindows(d,ventana[5]);
}

unven_cons() /* Para borrar los valores de los coeficientes*/
{ XUnmapWindow(d,constant[0]); XUnmapWindow(d,constant[1]); }

limpia_cons()
{
XCclearWindow(d,constant[0]);
XCclearWindow(d,constant[1]);
}

delay()
{ long tic;
for (tic=0;tic<100000;tic++);
}

Checa_evento() /*Se elige el tipo de polinomio a visualizar o para cambiar los valores de los coeficientes*/
{
l=0;
while(!l){
    XNextEvent(d,&evento);
    switch (evento.type)
    { case Expose:
        for(h=0;h<6;h++)
        {XSetForeground(d,grafind[h][0],white);
        XSetForeground(d,grafind[h][1],blue);
        XDrawString(d,indice[h],grafind[h][1],0,20,tipoctes[h+1],
        strlen(tipoctes[h+1]));
        }
        break;
        case ButtonPress:
        for (h=0;h<6;h++){
        if (evento.xany.window==indice[h])
        { XSetForeground(d,grafind[h][0],white);
        XSetForeground(d,grafind[h][1],red);
        XDrawString(d,indice[h],grafind[h][1],0,20,tipoctes[h+1],
        strlen(tipoctes[h+1]));j=h;
        indice1=j; cons=j; printf("\n h=%d \n",h);
        l=1;
        }
        }
        break;
    }
}
}

Hacer_ejes() /*Realiza los ejes de la función y del prisma*/
{
Ejes(d,ventana[1],grafic[1],75,vent[1].largo-50,vent[1].ancho,10);/*-50*/
Ejes(d,ventana[2],grafic[2],75,50,vent[2].ancho,vent[2].largo-30);
}

Hacer_matriz() /* Calcula valores de l efecto gravimétrico y se almacena en un archivo junot con "x" y "z" */

```

```

{out=fopen("ejemplo","wrb");
bandera=0;
n=101;
m=1;
for(i=1;i<=n;i++)
{ x0=-50+i-1;
  for(j=1;j<=m;j++)
  {
  z0=j-1;
  grav=glq(zi,xi,x0,z0,indice1,c);
  fprintf(out,"%5.3f%5.3f%5.3f\n",x0,z0,grav);
  ejem[i-1][0]=x0;
  ejem[i-1][1]=grav;
  }
}/*cerrar archivo*/
fclose(out);
}

Calcula_escala() /*Se obtiene el valor de la escala utilizada para el despliegue de la gráfica y del prisma*/
{float espX,espGx;
minx=ejem[0][0];
maxx=ejem[0][0];
mingx=ejem[0][1];
maxgx=ejem[0][1];
for(i=1;i<=n;i++)
{
  minx=menor(minx,ejem[i][0]);
  maxx=mayor(maxx,ejem[i][0]);
}
for(i=1;i<=n;i++)
{
  mingx=menor(mingx,ejem[i][1]);
  maxgx=mayor(maxgx,ejem[i][1]);
}
escala=escalar(minx,maxx,mingx,maxgx,500,300);
printf("mingx =%f maxgx=%f",mingx,maxgx);
xfijo=(int)((500-escala*(maxx-minx))/2);
printf("e: %f\n",escala);
}

Hacer_grafica() /*Dibuja la gráfica del efecto gravimétrico del prisma*/
{
Graffuncion(d,ventana[1],grafic[1],minx,maxx,mingx,maxgx,
escala,origen[0].x,origen[0].y,ejem,xfijo);
IntervaloZ(d,ventana[2],grafic[2],zi[1],zi[0],escala,
origen[1].x,origen[1].y);
IntervaloY(d,ventana[1],grafic[1],maxgx,mingx,escala,
origen[0].x,origen[0].y);
IntervaloX(d,ventana[1],grafic[1],maxx,minx,escala,origen[0].x,origen[0].y);
}

Hacer_rectangulo() /* Dibuja la figura del prisma en el despliegue*/
{int xin,xfi,zin,zfi;
float distxs,distzs;
distsx=xi[1]-xi[0];
distzs=zi[1]-zi[0];
xin=(int) -escala*(distsx/2);
xin=(int)(( 500-escala*distsx)/2);
xin=(int)(75+(500-escala*60)/2);
zin=50;
xfi=(int) xin+escala*distsx;
zfi=(int) zin+escala*distzs;
zfi=(int) 100;
XDrawLine(d,ventana[2],grafic[2],xin,zin,xfi,zin);
XDrawLine(d,ventana[2],grafic[2],xin,zin,xin,zfi);
XDrawLine(d,ventana[2],grafic[2],xin,zfi,xfi,zfi);
}

```

```

XDrawLine(d,ventana[2],grafic[2],xfi,zin,xfi,zfi);
XFillRectangle(d,ventana[2],grafic[2],xin,zin,xfi-xin,zfi-zin);
}

def_colors()
{ /*Lista de Colores usados en ambiente grafico*/
blue=MyColor(d,"blue");
gray=MyColor(d,"gray");
red=MyColor(d,"red");
beige=MyColor(d,"beige");
white=MyColor(d,"white");
black=MyColor(d,"black");

}
pon_etiquetas() /*Despliegue de etiquetas en las gráficas*/
{
XSetFont(d,grafic[1],f1);
XSetForeground(d,grafic[1],black);
XSetFont(d,grafic[2],f1);
XSetForeground(d,grafic[2],black);
XSetFont(d,grafic[5],f1);
XSetForeground(d,grafic[5],black);

XDrawString(d,ventana[1],grafic[1],13,120,etiq[1],strlen(etiq[1]));
XDrawString(d,ventana[1],grafic[1],13,140,etiq[2],strlen(etiq[2]));
XDrawString(d,ventana[1],grafic[1],13,160,etiq[3],strlen(etiq[3]));
XDrawString(d,ventana[1],grafic[1],13,180,etiq[4],strlen(etiq[4]));
XDrawString(d,ventana[1],grafic[1],13,200,etiq[5],strlen(etiq[5]));

XDrawString(d,ventana[2],grafic[2],250,25,etiq[0],strlen(etiq[0]));

XDrawString(d,ventana[2],grafic[2],13,30,etiq[6],strlen(etiq[6]));
XDrawString(d,ventana[2],grafic[2],13,50,etiq[7],strlen(etiq[7]));
XDrawString(d,ventana[2],grafic[2],13,70,etiq[8],strlen(etiq[8]));
XDrawString(d,ventana[2],grafic[2],13,90,etiq[9],strlen(etiq[9]));
XDrawString(d,ventana[2],grafic[2],13,110,etiq[10],strlen(etiq[10]));
XDrawString(d,ventana[2],grafic[2],13,130,etiq[11],strlen(etiq[11]));
XDrawString(d,ventana[2],grafic[2],13,150,etiq[12],strlen(etiq[12]));
XDrawString(d,ventana[2],grafic[2],13,170,etiq[13],strlen(etiq[13]));
XDrawString(d,ventana[2],grafic[2],13,190,etiq[14],strlen(etiq[14]));
XDrawString(d,ventana[2],grafic[2],13,210,etiq[15],strlen(etiq[15]));

XDrawString(d,ventana[5],grafic[5],75,25,etiq[16],strlen(etiq[16]));

XSetFont(d,grafic[1],f4);
XSetFont(d,grafic[2],f3);

XSetForeground(d,grafic[1],blue);
XSetForeground(d,grafic[2],blue);
XSetForeground(d,grafic[5],blue);

}

Inicio()
{
Hacer_ejes();
Hacer_matriz();
Calcula_escalas();
Hacer_grafica();
Hacer_rectangulo();
pon_etiquetas();
}

main()
{
int j=0;

```

```

xi[0]=-30; /*cambiar de -50 a -30*/
xi[1]=30;
zi[0]=0;
zi[1]=1; /*cambios de X oct 96cambiar de 1 a 2*/
grav=0;
cons=flag=0;
c[0]=2.3;
c[1]=1.2;
c[2]=-4;
c[3]=.82;
c[4]=.93;

/* Datos generadores de Ambiente XWindow */
d= XOpenDisplay(NULL); /*Display*/
r= RootWindow(d,0); /*Ventana Raiz*/
def_colors();
marco = 2; /*Medida del marco de ventanas en general*/
vent[0].x=100;vent[0].y=100;vent[0].ancho=1000;vent[0].largo=800;
vent[1].x=20; vent[1].y=20; vent[1].ancho=550; vent[1].largo=350;
vent[2].x=20; vent[2].y=400;vent[2].ancho=550; vent[2].largo=240;
vent[3].x=600;vent[3].y=50; vent[3].ancho=140; vent[3].largo=200;
vent[4].x=600;vent[4].y=300;vent[4].ancho=300; vent[4].largo=300;
vent[5].x=600;vent[5].y=300;vent[5].ancho=300; vent[5].largo=230;
vent[6].x=50; vent[6].y=90; vent[6].ancho=200; vent[6].largo=40;

origen[0].x=50;
origen[0].x=75;
origen[0].y=vent[1].largo-50;
origen[1].x=75;
origen[1].y=50;

ventana[0]=XCreateSimpleWindow(d,r,vent[0].x,vent[0].y,
vent[0].ancho,vent[0].largo,marco,blue,beige);
ventana[1]=XCreateSimpleWindow(d,ventana[0],vent[1].x,vent[1].y,
vent[1].ancho,vent[1].largo,marco,blue,white);
ventana[2]=XCreateSimpleWindow(d,ventana[0],vent[2].x,vent[2].y,
vent[2].ancho,vent[2].largo,marco,blue,white);
ventana[3]=XCreateSimpleWindow(d,ventana[0],vent[3].x,vent[3].y,
vent[3].ancho,vent[3].largo,marco,red,gray);
ventana[4]=XCreateSimpleWindow(d,ventana[0],vent[4].x,vent[4].y,
vent[4].ancho,vent[4].largo,marco,red,white);
ventana[5]=XCreateSimpleWindow(d,ventana[0],vent[5].x,vent[5].y,
vent[5].ancho,vent[5].largo,marco,blue,white);
ventana[6]=XCreateSimpleWindow(d,ventana[0],vent[6].x,vent[6].y,
vent[6].ancho,vent[6].largo,marco,blue,white);
for (i=0;i<5;i++){
opcion[i]=XCreateSimpleWindow(d,ventana[3],10,20+i*35,115,30,
marco,red,gray);
XSelectInput(d,opcion[i], ButtonPressMask|ExposureMask);
}
for (i=0;i<7;i++){
XSelectInput(d,ventana[i],ExposureMask | ButtonPressMask);
XMapWindow(d,ventana[i]);
grafic[i]=XCreateGC(d,ventana[i],0,0);
}
for (i=0;i<5;i++){
grafopc[i][0]=XCreateGC(d,opcion[i],0,0);
grafopc[i][1]=XCreateGC(d,opcion[i],0,0);
}
f1=XLoadFont(d,"timB114");
f2=XLoadFont(d,"timB112");
f3=XLoadFont(d,"timB12");
f4=XLoadFont(d,"courO12");
for (i=0;i<5;i++){
XSetFont(d,grafopc[i][0],f1);
XSetFont(d,grafopc[i][1],f1);
}

```

```

XSetFont(d,grafic[1],f4);
menu2=XCreateSimpleWindow(d,ventana[0],750,50,140,240,3,red,gray);
XMapWindow(d,menu2);
XSelectInput(d,menu2,ExposureMask);
grafmen=XCreateGC(d,menu2,0,0);
for (i=0;i<6;i++) {
    indice[i]=XCreateSimpleWindow(d,menu2,10,20+i*35,115,30,
        marco,red,white);
    XSelectInput(d,indice[i],ExposureMask|ButtonPressMask);
    grafind[i][0]=XCreateGC(d,indice[i],0,0);
    grafind[i][1]=XCreateGC(d,indice[i],0,0);
}
XMapSubwindows(d,menu2);
for(i=0;i<6;i++)
    {
        XSetForeground(d,grafind[i][0],white);
        XSetForeground(d,grafind[i][1],blue);
        XSetFont(d,grafind[i][0],f3);
        XSetFont(d,grafind[i][1],f3);
    }
XUnmapWindow(d,ventana[4]);
XMapWindow(d,ventana[5]);
XMapSubwindows(d,ventana[3]);
XSelectInput(d,ventana[5],ExposureMask);
XSelectInput(d,ventana[6],ButtonPressMask);
XMapSubwindows(d,ventana[5]);
XSetForeground(d,grafic[6],blue);
XSetForeground(d,grafic[5],blue);
PonDensidad(d,ventana[5],grafic[5]);
PonLinDen(d,ventana[6],grafic[6]);

constant[0]=XCreateSimpleWindow(d,ventana[5],50,150,60,50,3,blue,white);
constant[1]=XCreateSimpleWindow(d,ventana[5],150,150,110,50,3,blue,white);
XSelectInput(d,constant[0],ExposureMask);
XSelectInput(d,constant[1],ExposureMask);
grafcons[0]=XCreateGC(d,constant[0],0,0);
grafcons[1]=XCreateGC(d,constant[1],0,0);
XSetForeground(d,grafcons[0],blue);
XSetForeground(d,grafcons[1],blue);

while(1)
    {
        XNextEvent(d,&evento);
        switch (evento.type)
        { case Expose:
            for(i=0;i<3;i++)
                XSetForeground(d,grafic[i],blue);
            for(i=0;i<5;i++)
                {XSetForeground(d,grafopc[i][0],white);
                XSetForeground(d,grafopc[i][1],blue);
                PonOpcion(d,opcion[i],grafopc[i][0],
                grafopc[i][1]);
            }
            menu_2();
            Inicio();
            break;
            case ButtonPress:
            for (i=0;i<5;i++)
            {
                if (evento.xany.window==opcion[i])
                    { XSetForeground(d,grafopc[i][0],blue);
                    XSetForeground(d,grafopc[i][1],white);
                    PonOpcion(d,opcion[i],grafopc[i][0],grafopc[i][1]); j=i;
                    }
            }
            switch(j)

```



```

        return(gx);
    }

float maximo(x1,x2)
float x1,x2;
{float temp,fx,valor;
  temp=x1;
  valor=funcion(temp);
  while (temp<=x2)
  {
    fx=funcion(temp);
    if (fx>valor) valor=fx;
    temp+=0.5;
  }
return(valor);
}

float minimo(x1,x2)
float x1,x2;
{float temp,fx,valor;
  temp=x1;
  valor=funcion(temp);
  while (temp<x2)
  {
    fx=funcion(temp);
    if (fx<valor) valor=fx;
    temp+=0.5;
  }
return(valor);
}

float menor(x1,x2)
float x1,x2;
{float vmenor;
  if (x1<x2) vmenor=x1;
  else      vmenor=x2;
return(vmenor);
}

float escalar(x1,x2, fx1, fx2, ancho, largo)
float x1,x2,fx1,fx2;
int ancho,largo;
{
  float distanciax,distanciafx,escx,escfx,escala;
  distanciax=x2-x1;
  distanciax+=distanciax*0.1;
  distanciafx= (float) fx2-fx1;
  distanciafx+=distanciafx*0.1;
  escx=ancho/distanciax;
  escfx=largo/distanciafx;
  escala=menor(escx,escfx);
return(escala);
}

void GrafFuncion(d,w,gc,xmin,xmax,fxmin,fxmax,esc,coordx,coordy)
Display *d;
Window w;
GC gc;
float xmin,xmax,fxmin,fxmax,esc;
int coordx,coordy;
{
  float xi,fxi,tempfx=0,tempx=0,tempcen;
  int intx1,intfx1,intx2,intfx2,bandera;

  bandera=0;
  tempfx=-1*(fxmin*esc);

```

```

if (xmin<0) tempx=-1*(xmin*esc);
tempcen=(500.0-esc*(xmax-xmin))/2.0;
xi=xmin;
while(xi<=xmax)
{
  fxi=funcion(xi);
  intx1=(int)(coordx +tempx+esc*xi)+(int)tempcen;
  intfx1=(int)(coordx -(tempfx+ esc*fxi));
  if( bandera==1)
  {
    XDrawLine(d,w,gc,intx2,intfx2,intx1,intfx1);
    intx2=intx1;
    intfx2=intfx1;
  }
  else
  {
    intx2=intx1;
    intfx2=intfx1;
    bandera=1;
  }
  xi+=.5;
}
}

void Esfera(d,w,gc,esc,color)
Display *d;
Window w;
GC gc;
char *color;
float esc;
{
  int centro_esf,esfe1,esfe2, esfe3;
  centro_esf=250;
  esfe1=(int) centro_esf*esc*rad;
  esfe2=(int) 2*esc*rad;
  esfe3=(int) esc*(z-rad);
  XSetForeground(d,gc,color);
  XDrawArc(d,w,gc,esfe1,esfe3,esfe2,esfe2,0,360*64);
  XFillArc(d,w,gc,esfe1,esfe3,esfe2,esfe2,0,360*64);
}

void PonOpcion(d,w,gc1,gc2)
Display *d;
Window w;
GC gc1,gc2;
{
  XFillRectangle(d,w,gc1,0,0,120,30);
  XDrawString(d,w,gc2,0,20,opciones[i],strlen(opciones[i]));
}
itoa(n,s)
char s[];
int n;
{
  int i,signo;

  if((signo=n)<0) n=-n;
  i=0;
  do { s[i++]=n % 10 +'0';
    } while ((n/=10)>0);
  if (signo<0) s[i++]='-';
    s[i]='\0';

  reverse(s);
}
reverse(s)
char s[];
{
  int c,i,j;
  for (i=0,j=strlen(s)-1;i<j;i++,j--)
  {
    c=s[i];

```

```

        s[i]=s[j];
        s[j]=c;
    }
}
void PonDensidad(d,w,gc)
Display *d;
Window w;
GC gc;
{
    char *cad[1];
    int x0,y0,x1,y1,k;
    cad[0]="3";
    y0=90;
    y1=130;
    k=-3;
    XSetLineAttributes(d,gc,1,LineSolid,CapButt,JoinBevel);
    XDrawLine(d,w,gc,60,70,240,70);
    for(i=60;i<=240;i+=30,k++)
        { XDrawLine(d,w,gc,i,65,i,75);
          itoa(k,cad[0]);
          XDrawString(d,w,gc,i-2,60,cad[0],strlen(cad[0]));
        }
}

void PonLinDen(d,w,gc)
Display *d;
Window w;
GC gc;
{
    int dens,x1,y0,y1;
    XDrawLine(d,w,gc,0,20,200,20);
    if (band==1)
        {x1=158;
         y0=0;
         y1=40;
         XDrawLine(d,w,gc,x1,y0,x1,y1);
        }
    band=0;
}

void CambiaZ(d,w,gc,esc)
Display *d;
Window w;
GC gc;
float esc;
{
    XEvent profundidad;
    int y0,y1,x0,x1;
    x0=(int) centro;
    y0=0;
    XNextEvent(d,&profundidad);
    if (profundidad.xany.window==w)
        switch(profundidad.type)
            {
                case ButtonPress:
                    y1=profundidad.xbutton.y;
                    XDrawLine(d,w,gc,x0,y0,x0,y1);
                    z=(float)(1/esc)*y1;
                    if (z<rad) z=rad;
                    printf("nuevo valor de z= %f\n",z);
                    break;
            }
}

void CambiaRadio(d,w,gc,esc)
Display *d;
Window w;

```

```

GC gc;
float esc;
{
XEvent radio;
int x0,y0,x1,y1;
x0= (int) centro;
y0= esc*z;
XNextEvent(d,&radio);
if (radio.xany.window==w)
switch(radio.type)
{
case ButtonPress:
x1=radio.xbutton.x;
y1=radio.xbutton.y;
rad=(float) ((float) (y0-y1)/(float)esc);
if (rad<0) rad*=-1;
if (rad>50.0) rad=50;
if (rad>z) z=rad;
printf("radio=%f\n prof=%f\n",rad,z);
break;
}
}
void CambiaDensidad(d,w,gc)
Display *d;
Window w;
GC gc;
{
XEvent densidad;
int x1,y1,y0;
y0=0;
y1=40;
while(1)
{
XNextEvent(d,&densidad);
if (densidad.xany.window==w)
switch(densidad.type)
{
case ButtonPress:
x1=densidad.xbutton.x;
break;
case ButtonRelease:
x1=densidad.xbutton.x;
den=x1*0.033333-3.0;
break;
}
XClearWindow(d,w);
XDrawLine(d,w,gc,x1,y0,x1,y1);
XDrawLine(d,w,gc,0,20,200,20);
printf("x1 densid=%d\n",x1);
if (densidad.xbutton.button==Button3) break;
}
}

void IntervaloY(d,w,gc,max,min,esc,origenx,origeny)
Display *d;
Window w;
GC gc;
float max,min,esc;
int origenx,origeny;
{
float distancia;
int primera,segunda,contador,gx_entera,num_intervalos,num_graf;
char *cad[1];
cad[0]="3";
contador=0;
primera=origenx-5;

```

```

segunda=origenx;
distancia= esc*(max-min);
num_intervalos=(int) (distancia/10);
gx_entera=origeny;
num_graf=(int) min;
while (contador<=num_intervalos+1)
{
  XDrawLine(d,w,gc,primera,gx_entera,segunda,gx_entera);
  itoa(num_graf,cad[0]);
  XDrawString(d,w,gc,segunda-40,gx_entera+2,cad[0],strlen(cad[0]));
  num_graf+=(int) (10/esc);
  gx_entera-=10;
  contador++;
}
}
void IntervaloZ(d,w,gc,max,esc,origenx,origeny)
Display *d;
Window w;
GC gc;
float max,esc;
int origenx,origeny;
{
  int distancia,min;
  int primera,segunda,contador,gx_entera,num_intervalos,num_graf;
  char *cad[1];
  cad[0]="3";
  primera=origenx-5;
  segunda=origenx;
  distancia=(int) esc*z;
  contador=0;
  min=0;
  num_intervalos=2;
  gx_entera=origeny;
  num_graf=0;
  while (contador< num_intervalos)
  {
    XDrawLine(d,w,gc,primera,gx_entera,segunda,gx_entera);
    itoa(num_graf,cad[0]);
    XDrawString(d,w,gc,segunda-40,gx_entera+2,cad[0],strlen(cad[0]));
    num_graf=(int)z;
    printf("%f esc\n",esc);
    gx_entera+=distancia;
    contador++;
  }
}
void IntervaloX(d,w,gc,max,min,esc,origenx,origeny)
/*vent[2].x=100; vent[2].y=450;*/
Display *d;
Window w;
GC gc;
float max,min,esc;
int origenx,origeny;
{
  float distancia;
  int primera,segunda,contador,gx_entera,num_intervalos,num_graf,uno,dos,temp;
  char *cad[1];
  cad[0]="3";
  contador=0;
  primera=origeny;
  segunda=origeny+10;
  distancia= esc*(max-min);
  temp=(int)((500.0-distancia)/2);
  origenx+=temp;
  gx_entera=origenx;

  if (distancia>400) num_intervalos=20;

```

```

else if (distancia < 200 && distancia >= 100) num_intervalos=4;
    else if (distancia < 100) num_intervalos=2;
    else num_intervalos=10;

uno=(int)((max-min)/num_intervalos);
dos=(int)((500-2*temp)/num_intervalos);
num_graf=(int)min;
while (contador<=num_intervalos)
{
    XDrawLine(d,w,gc,gx_entera,primera,gx_entera,segunda);
    itoa(num_graf,cad[0]);
    XDrawString(d,w,gc,gx_entera-4,origeny+25,cad[0],strlen(cad[0]));
    num_graf+=(int)uno;
    gx_entera+=(int)dos;
    contador++;
}
}

```

*Fin de esfera.h*

*Inicio de esfera.c*

```

#include <X11/Xlib.h> /*Encabezados de X Window necesarios*/
#include <X11/Xutil.h>
#include <stdio.h>
#include "esfera.h"
#define marco 3

Display *d;
Window r,ventana[6],opcion[5],densidad[5],menu2,index2[6];
GC grafic[6],grafopc[5][2],grafdensidad,grafmen,grafind[6][2];
char *etiqa[8]={"k i l o m e t r o s","m","G","a","l","s","D E N S I D A D","k i l o m e t r o s"};
XEvent evento;
Font f1,f2,f3;
unsigned long MyColor();
unsigned long blue,red,gray,white,black,brown,beige;
XSetWindowAttributes att;

static struct {
    int x,y,largo,ancho;
} vent[6];
static struct {
    int x,y;
} origen[3];

float x,escala;
float minx,maxx,mingx,maxgx;

IniciaVentanas()
{
    maxx=maximo(minx,maxx);
    mingx=minimo(minx,maxx);
    escala=escalar(minx,maxx,(float) mingx, (float) maxx, vent[1].ancho,vent[1].largo);

    XClearWindow(d,ventana[0]);
    XClearWindow(d,ventana[1]);
    XClearWindow(d,ventana[2]);
    XClearWindow(d,ventana[3]);
    XClearWindow(d,ventana[4]);

    for(i=0;i<5;i++)
    {XSetForeground(d,grafopc[i][0],white);
      XSetForeground(d,grafopc[i][1],blue);
      PonOpcion(d,opcion[i],grafopc[i][0],grafopc[i][1]);
    }
    XSetForeground(d,grafic[5],blue);
    PonDensidad(d,ventana[4],grafic[4]);

```

```

}
ReFuncion()
{
  Ejes(d,ventana[0],grafic[0],vent[1].x,vent[1].y,
      vent[1].ancho+vent[1].x,vent[1].largo+vent[1].y);
  GrafFuncion(d,ventana[1],grafic[1],minx,maxx,mingx,maxgx,
      escala,origen[1].x,origen[1].y);
  Esfera(d,ventana[2],grafic[2],escala,blue);
  PonLinDen(d,ventana[5],grafic[5]);
  IntervaloY(d,ventana[0],grafic[0],maxgx,mingx,escala,
      origen[0].x,origen[0].y);
  IntervaloZ(d,ventana[0],grafic[0],z,escala,
      vent[2].x,vent[2].y);
  IntervaloX(d,ventana[0],grafic[0],maxx,minx,escala,
      origen[0].x,origen[0].y);
  pon_etiquetas();
}

def_colors()
{
  /*Lista de Colores usados en ambiente grafico*/
  blue=MyColor(d,"blue");
  gray=MyColor(d,"gray");
  red=MyColor(d,"red");
  beige=MyColor(d,"beige");
  white=MyColor(d,"white");
  black=MyColor(d,"black");
  brown=MyColor(d,"brown");
}

pon_etiquetas()
{
  XSetFont(d,grafic[0],f1);
  XSetForeground(d,grafic[0],black);
  XSetFont(d,grafic[4],f1);
  XSetForeground(d,grafic[4],black);

  XDrawString(d,ventana[0],grafic[0],30,120,etiq[1],strlen(etiq[1]));
  XDrawString(d,ventana[0],grafic[0],30,140,etiq[2],strlen(etiq[2]));
  XDrawString(d,ventana[0],grafic[0],30,160,etiq[3],strlen(etiq[3]));
  XDrawString(d,ventana[0],grafic[0],30,180,etiq[4],strlen(etiq[4]));
  XDrawString(d,ventana[0],grafic[0],30,200,etiq[5],strlen(etiq[5]));
  XDrawString(d,ventana[0],grafic[0],270,420,etiq[7],strlen(etiq[7]));
  XDrawString(d,ventana[4],grafic[4],107,25,etiq[6],strlen(etiq[6]));

  XSetFont(d,grafic[0],f2);
  XSetForeground(d,grafic[0],blue);
  XSetForeground(d,grafic[4],blue);
}

main()
{
  int j=0;
  /* Datos generadores de Ambiente XWindow */
  /*Display*/
  d=XOpenDisplay(NULL);
  /*Ventana Principal*/
  r=RootWindow(d,0);
  def_colors();
  vent[0].x=100; vent[0].y=100; vent[0].ancho=1000; vent[0].largo=800;
  vent[1].x=100; vent[1].y=50; vent[1].ancho=500; vent[1].largo=300;
  vent[2].x=100; vent[2].y=450; vent[2].ancho=500; vent[2].largo=300;
  vent[3].x=650; vent[3].y=50; vent[3].ancho=150; vent[3].largo=200;

```

```

vent[4].x=650; vent[4].y=300; vent[4].ancho=300; vent[4].largo=200;
vent[5].x=50; vent[5].y=90; vent[5].ancho=200; vent[5].largo=40;

origen[0].x=vent[1].x;
origen[0].y=vent[1].y+vent[1].largo;
origen[1].x=0;
origen[1].y=vent[1].largo;

ventana[0]=XCreateSimpleWindow(d,r,vent[0].x,vent[0].y,
    vent[0].ancho,vent[0].largo,marco,red,beige);
ventana[1]=XCreateSimpleWindow(d,ventana[0],vent[1].x,vent[1].y,
    vent[1].ancho,vent[1].largo,marco,red,white);
ventana[2]=XCreateSimpleWindow(d,ventana[0],vent[2].x,vent[2].y,
    vent[2].ancho,vent[2].largo,marco,red,gray);
ventana[3]=XCreateSimpleWindow(d,ventana[0],vent[3].x,vent[3].y,
    vent[3].ancho,vent[3].largo,marco,blue,gray);
ventana[4]=XCreateSimpleWindow(d,ventana[0],vent[4].x,vent[4].y,
    vent[4].ancho,vent[4].largo,marco,blue,white);
ventana[5]=XCreateSimpleWindow(d,ventana[4],vent[5].x,vent[5].y,
    vent[5].ancho,vent[5].largo,marco,blue,white);

for (i=0;i<5;i++) {
    opcion[i]=XCreateSimpleWindow(d,ventana[3],10,20+i*35,120,30,
        marco,red,gray);
    XSelectInput(d,opcion[i],ExposureMask |ButtonReleaseMask
        |ButtonPressMask);
}
for (i=0;i<6;i++){
    XSelectInput(d,ventana[i],ExposureMask | ButtonPressMask);
    XMapWindow(d,ventana[i]);
    grafic[i]=XCreateGC(d,ventana[i],0,0);
    if (i<5)
    {
        grafopc[i][0]=XCreateGC(d,opcion[i],0,0);
        grafopc[i][1]=XCreateGC(d,opcion[i],0,0);
    }
}

f1=XLoadFont(d,"timBI14");
f2=XLoadFont(d,"timBI12");

for (i=0;i<5;i++) {
    XSetFont(d,grafopc[i][0],f1);
    XSetFont(d,grafopc[i][1],f1);
}

XSelectInput(d,ventana[4],ButtonPressMask|ButtonReleaseMask
    |ButtonMotionMask|EnterWindowMask|ExposureMask);
XSelectInput(d,ventana[5],ButtonPressMask|ButtonReleaseMask
    |ButtonMotionMask|EnterWindowMask|ExposureMask);
XMapSubwindows(d,ventana[3]);
XMapSubwindows(d,ventana[4]);
minx=-50;
maxx=50;
maxgx=maximo(minx,maxx);
mingx=minimo(minx,maxx);
escala=escalar(minx,maxx,(float) mingx, (float) maxgx, vent[1].ancho,vent[1].largo);
printf("escala =%f\n",escala);
while(1)
{
    XNextEvent(d,&evento);
    switch (evento.type)
    { case Expose:
        for(i=0;i<2;i++)
            XSetForeground(d,grafic[i],blue);
        for(i=0;i<5;i++)

```

