



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

"Desarrollo de un sistema para el manejo
de los datos del TNS de Sierra Negra."

T E S I S

QUE PARA OBTENER EL TÍTULO DE :

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

DANIEL RODRÍGUEZ OSORIO



ASESOR:

DR. JOSÉ FRANCISCO VALDÉS GALICIA.

Febrero 2009



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos.

A mi madre, por más de un millón de razones, simplemente por estar siempre ahí... con la palabra exacta, siempre con entrega para cumplir cabalmente con esa labor de madre, por su adecuada guía y por su profesionalismo demostrado día a día en los 28 años que tengo de conocerla y observar su desempeño como la mejor mamá del mundo.

A mi padre, el señor Jesús Rodríguez Vega, por ser el sustento del hogar en el que crecí, por todo su esfuerzo y entereza, por ser uno de los pilares de este largo trayecto académico, por enseñarme sin palabras, el valor de la vida, el trabajo y la vocación de ser padre.

A mi esposa Yadira, por su amor, apoyo y comprensión incondicionales, por su compañía y por haberme dado el regalo más bello que he recibido, mi pequeño Daniel.

Al Dr. José Francisco Valdés Galicia por su confianza y apoyo, por su tutoría en el presente trabajo, por ser parte importante para la culminación de mis estudios profesionales al llevar la dirección de esta tesis.

A mis compañeros Alejandro, Sergio y Carlos, por haberle dado un nuevo significado a mi estancia en la facultad de ingeniería.

A todas las personas que he conocido a lo largo de mi vida, por que de todas y cada una de ellas he recibido una enseñanza distinta.

Índice general.

Introducción.....	XI
Planteamiento del problema.....	XII
Capítulo 1. El Sol, actividad solar y emisiones solares.....	1
1.1 Importancia del estudio del Sol.....	2
1.2 Formación del sistema solar.....	2
1.3 El Sol, su estructura y actividad solar.....	3
1.3.1 Descripción general del Sol.....	3
1.3.2 Estructura del Sol, producción y transporte de energía.....	4
1.3.2.1 Región Interna.....	4
1.3.2.1.1 El Núcleo.....	4
1.3.2.1.2 La zona radiactiva.....	6
1.3.2.1.3 La zona convectiva.....	6
1.3.2.2 La Atmósfera Solar.....	7
1.3.2.2.1 La Fotosfera.....	7
1.3.2.2.2 La Cromosfera.....	8
1.3.2.2.3 La Corona.....	8
1.3.3 Actividad solar.....	9
1.3.3.1 La rotación diferencial del Sol.....	9
1.3.3.2 El campo magnético del Sol.....	10
1.3.3.3 Manchas solares.....	10
1.3.3.4 Protuberancias.....	11

1.3.3.5	Filamentos.....	11
1.3.3.6	Playas.....	11
1.3.3.7	Fulguraciones.....	12
1.3.3.8	Eyecciones de masa coronal.....	13
1.3.3.9	Viento solar.....	14
1.3.3.10	Emisiones solares.....	15
1.3.3.11	Detección de partículas cargadas.....	16
Capítulo 2. Herramientas de software.....		19
2.1	El software libre.....	20
2.2	El sistema operativo con núcleo Linux.....	21
2.3	Paradigma de programación orientada a objetos.....	22
2.3.1	Características esenciales de la programación orientada a objetos.....	22
2.4	Diseño de sistemas con UML.....	24
2.5	Lenguaje Java.....	26
2.5.1	Descripción del lenguaje de programación orientada a objetos: Java.....	26
2.5.2	Estructura de Java.....	28
2.5.3	El entorno de aplicaciones Java.....	29
Capítulo 3. El Telescopio de Neutrones Solares de Sierra Negra, Puebla y Diseño del sistema.....		31
3.1	El Telescopio de Neutrones Solares (TNS) de Sierra Negra, Puebla.....	32
3.1.1	Antecedentes.....	32
3.1.2	El Telescopio de Neutrones Solares.....	33

3.1.3	El sistema de cómputo.....	37
3.2	Diseño del sistema.....	39
3.2.1	Análisis de requerimientos.....	39
3.2.2	Diseño del sistema con UML.....	41
3.2.2.1	Diagrama de casos de uso.....	41
3.2.2.2	Diagrama de paquetes.....	43
3.2.2.3	Diagrama de clases.....	43
3.2.2.4	Diagrama de actividades.....	44
3.2.2.5	Diagrama de secuencia.	44
3.3	Conclusiones del diseño.....	45
 Capítulo 4. El Sistema: Programación, estructura y funcionamiento.....		55
4.1	Descripción general de la estructura del sistema.....	56
4.2	Descripción general del funcionamiento del sistema.....	57
4.2.1	Ejecución del sistema.....	57
4.2.2	Ventana principal.....	57
4.2.3	El menú de opciones.....	59
4.2.4	Ventanas secundarias.....	60
4.2.4.1	Ventana para salvar un archivo o gráfica.....	60
4.2.4.2	Ventana para solicitar una gráfica de dos días.....	60
4.2.2.3	Ventana para solicitar una gráfica de otros canales.....	61
4.3	Descripción del funcionamiento del sistema a nivel de subprogramas u objetos.....	61
4.4	Descripción de las clases empleadas y sus métodos.....	63
4.4.1	Clases del paquete IG.....	63
4.4.2	Clases del paquete Datos.....	65

4.4.3	El paquete Graficas.....	70
4.4.4	Utilerias.....	70
4.4.5	Parametros.....	71
4.5	Salidas del sistema.....	71
Capítulo 5.	Uso del sistema sngraphics.....	75
5.1	Requerimientos del sistema (a nivel software y hardware).....	76
5.2	Manual de usuario.....	77
5.2.1	Descripción general.....	77
5.2.2	Pantalla principal del sistema.....	79
5.2.3	Gráficas.....	80
5.2.4	Tablas.....	82
5.2.5	Menú de opciones.....	82
5.2.5.1	Archivo.....	82
5.2.5.1.1	Salvar 4 canales.....	82
5.2.5.1.2	Salvar canales.....	83
5.2.5.1.3	Salvar directorio.....	84
5.2.5.1.4	Mostrar 4 canales.....	84
5.2.5.1.5	Mostrar canales (todos los canales).....	85
5.2.5.2	Graficación.....	86
5.2.5.2.1	Graficar 2 días.....	86
5.2.5.2.2	Graficar otros canales.....	86
5.2.5.2.3	Salvar gráfica como: .tex, .ps ó .png	88
5.2.5.3	Borrar.....	89

5.2.5.3.1 Borrar temporales.....	89
5.2.5.4 Acerca de.....	89
5.2.5.4.1 Acerca de sngraphics.....	89
5.2.5.5 Ayuda.....	89
5.2.5.5.1 Ayuda en html.....	89
5.3 Conclusiones.....	90
Apéndice A. Manual de instalación.....	93
Glosario de términos.....	97
Bibliografía.....	100

Introducción.

En la superficie del Sol se generan procesos de liberación súbita de energía (fulguraciones y eyecciones de masa coronal), los cuales emiten señales electromagnéticas, partículas cargadas y neutrones; algunas de estas emisiones pueden ser captadas en la Tierra por medio de observatorios con equipos diseñados para este propósito. El Telescopio de Neutrones Solares (TNS) ubicado en Sierra Negra, Puebla, tiene el objetivo de captar las emisiones de neutrones que tienen su origen en estos procesos. Para ello se utiliza un sistema de detección bastante ingenioso que consiste en 4 góndolas que en su base contienen plásticos centelladores y, en la parte superior, fototubos multiplicadores, las góndolas están rodeadas por capas con contadores proporcionales, formando un así un cubo que es protegido a los lados por 4 paredes de hierro y en la parte superior por una capa de acero. Posteriormente, el módulo antes descrito, se conecta electrónicamente a un dispositivo (CAMAC) que se encarga de interpretar las detecciones de partículas realizadas, este módulo a su vez se conecta a un equipo de cómputo en el cual se archivan los datos correspondientes a las detecciones realizadas, en archivos con formato ASCII.

Es ahí de donde parte el objetivo de este trabajo, el desarrollo de un software, para el manejo de los datos que genera el TNS. Debido a esto, lo primero que se contempla en este trabajo, es un entendimiento de la estructura, el transporte de energía y la emisión de partículas que se da en la estrella de nuestro Sistema Planetario (el Sol).

Posteriormente describo las herramientas de software que se utilizaron para el desarrollo del sistema requerido; se decidió trabajar bajo plataforma Linux, elemento indispensable para el desarrollo, de acuerdo a lo solicitado por el Dr. José Francisco Valdés Galicia, quien dirigió este trabajo de tesis, y además es el responsable del TNS de Sierra Negra. Junto con él, tomé la decisión de los demás elementos a utilizar: UML, para el diseño del sistema, JAVA para la programación del mismo y GNPlot para la realización de las gráficas.

Lo siguiente es entender el funcionamiento del Telescopio de Neutrones Solares y su sistema de cómputo, con el propósito de conocer la estructura de los archivos generados por el TNS, posteriormente se analizaron los requerimientos para la construcción del sistema, y con ello, se creó el diseño con diagramas estáticos y dinámicos, que cumplen con el estándar del UML, con el propósito de describir la estructura y el funcionamiento del software a desarrollar.

Por último, el presente trabajo, muestra un desglose de los elementos que conforman todo el sistema desarrollado y una explicación de cómo usarlo y como instarlo.

Planteamiento del problema.

Se requiere diseñar e implementar un sistema que procese, manipule y grafique los datos que se han obtenido, y se siguen obteniendo del Telescopio de Neutrones Solares de Sierra Negra, Puebla.

Los datos que se tienen actualmente se encuentran en dos formatos diferentes debido a la actualización del sistema de cómputo, por lo cual, para datos anteriores al 2005 se trabajará con archivos de las siguientes características:

1. Fueron generados por un sistema UNIX Solaris.
2. Tienen un formato de texto basado en codificación ASCII.
3. Están comprimidos con el compresor gzip de UNIX Solaris.
4. Tienen extensión .sn*, donde * indica el número de archivo consecutivo generado en un día.
5. Presentan (ya descomprimidos), una distribución de 65 columnas y 8,640 filas (en total del muestreo para un día entero), en donde cada fila corresponde al muestreo de cada 10 segundos, que realiza el TNS.

El segundo formato de datos sobre el que se trabajará tiene las siguientes características:

1. Son generados por un sistema Linux Fedora Core 1.
2. Tienen un formato de texto basado en codificación ASCII.
3. Los archivos ya no se encuentran comprimidos
4. Tienen extensión .sn*, donde * indica el número de archivo consecutivo generado en un día.
5. Presentan una distribución de 54 columnas por 8645 filas, en donde las primeras 5 filas solo tiene una columna y son los datos de algunos parámetros de configuración, posteriormente viene los datos propiamente hablando.

Los datos se obtienen por diversas vías: medio de almacenamiento (CD, Disquete, Disco Duro, etc.). El usuario requiere gráficas o tablas de los datos con las siguientes resoluciones:

1. 1 minuto
2. 3 minutos
3. 5 minutos
4. 1 hora

El archivo original almacena datos con resolución de 10 segundos.

Las resoluciones requeridas resultan de la acumulación de los de menor resolución, por ejemplo para hacer el cálculo del dato correspondiente al primer minuto del día, se deberán sumar los 6 primeros datos de 10 segundos, para generar el primer dato del día de 3 minutos se deberán sumar los 3 primeros datos de 1 minuto, y así sucesivamente; este procedimiento aplica para todos los

canales de datos, exceptuando los referentes a presión y temperatura, ya que para estos los cálculos deberán hacerse obteniendo promedios.

Esto permitirá delimitar la información que al usuario le interese en un determinado momento, teniendo como punto central el poder visualizar una gráfica o tabla de los datos de interés.

Método.

El sistema requerido puede diseñarse e implementarse de diversas formas sobre varias plataformas y utilizar diversos lenguajes y paradigmas de programación y metodologías de diseño, para el diseño del sistema existen metodologías diversas, por ejemplo UML u OMT, que son herramientas de diseño de software probadas y empatadas con la metodología de programación orientada a objetos, esto nos llevaría a decidir utilizar un lenguaje de programación que se base en el paradigma orientado a objetos.

Revisando bibliografía y sistemas desarrollados, la mayoría se diseñaron con UML, demostrando eficacia y facilidad en el diseño, por ello utilizaré este método de diseño.

Una cuestión sumamente importante es el sistema operativo sobre el cual se desarrollará, inicialmente pensaría en utilizar windows, dado que es un sistema de dominio publico y mucho más familiar, además existen compiladores de diversos lenguajes que cumplen con el paradigma orientado a objetos, y cuya funcionalidad está probada, por ejemplo Visual Basic, Java, Visual Java, Visual C++, entre muchos otros, el gran inconveniente es el pago de una licencia de software, mi director de tesis me planteo la necesidad de utilizar el sistema operativo con núcleo Linux, y Java como lenguaje de programación; el diseño con UML y la programación en Java tienen un vínculo muy estrecho.

Una vez analizadas las opciones, la conclusión es la siguiente:

- Se diseñará el sistema con UML(Unified Modeling Language, lenguaje de modelado unificado).
- Se trabajará sobre el sistema operativo Linux, específicamente con la distribución, fedora, en su versión 1.0.
- Se utilizará Java como lenguaje de programación.
- Se ocupará GNUPlot para la graficación de los datos.

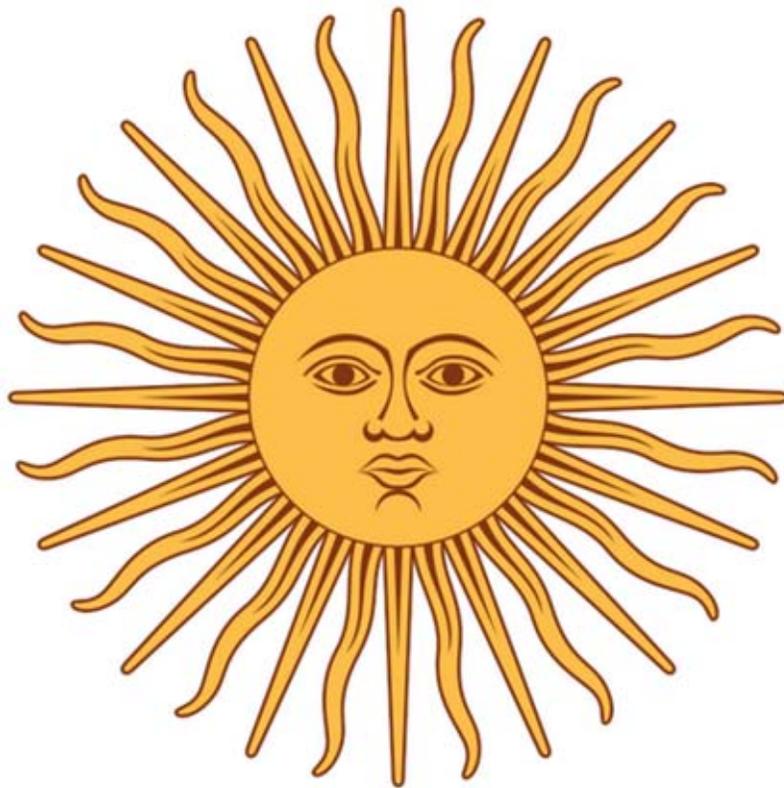
Estás fueron las herramientas seleccionadas, pero los pasos de desarrollo del presente trabajo se seguirán de acuerdo al siguiente listado:

1. Recopilación de información para ubicar el contexto del sistema.

- Información acerca del Sol, actividad solar y sus emisiones.
2. Recopilación de información para ubicar el marco teórico del desarrollo del software, desarrollo de sistemas, diseño y programación con UML y java.
 3. Análisis de requerimientos del sistema.
 4. Diseño del sistema con UML
 5. Generación de código (Java y GNUPlot)
 6. Pruebas al sistema

Capítulo 1.

El Sol, actividad solar y emisiones solares.



2 Capítulo 1. El Sol, actividad solar y emisiones solares

Para poder contextualizar nuestro desarrollo de software, es necesario conocer la estructura y comportamiento de nuestro Sol, y como se lleva a cabo la producción de energía y sobre todo la emisión de partículas que llegan a la Tierra y son detectadas por el TNS, en Sierra Negra. Es por ello que el presente capítulo de esta tesis se refiere por completo a la estrella de nuestro Sistema Planetario.

1.1 Importancia del estudio del Sol.

El Sol es la estrella más cercana a la Tierra, es la fuente primordial de vida en nuestro planeta, la energía que despide y en las formas y magnitudes que nos llegan son las ideales para el desarrollo y mantenimiento de la vida. Por otra parte, desde el punto de vista científico, el Sol, es un área de estudio imprescindible para el desarrollo de las ciencias físicas, porque es ahí donde ocurren una cantidad impresionante de fenómenos que solamente tienen lugar en las condiciones específicas que reinan en el Sol, es un instrumento de estudio importante para entender el comportamiento de gran parte del Universo, y sobre todo de nuestro Sistema Planetario.

Existen fenómenos en el Sol que el ser humano estudia pero aún no ha podido explicar, por ejemplo, el campo magnético solar; en contraste hay fenómenos, que si se han podido explicar, y que han sido fundamentales en el desarrollo de la física nuclear y otras áreas, un ejemplo de estos fenómenos es la fusión nuclear que ocurre en el núcleo del Sol y que es la fuente principal de energía de la estrella de nuestro Sistema Planetario.

Otro aspecto fundamental que nos dice porque hay que estudiar los fenómenos que ocurren en el Sol es el futuro aprovechamiento de la energía solar, hoy en día existen instrumentos que captan la energía del Sol para convertirla en energía eléctrica, por ejemplo, pero son dispositivos basados en celdas solares cuyas dimensiones son muy grandes y producen poca energía, se espera que con el estudio y entendimiento de los fenómenos que ocurren en el Sol se pueda optimizar el aprovechamiento de la energía solar.

1.2 Formación del sistema solar.

Hace aproximadamente cinco mil millones de años, debido al colapso gravitatorio de una nube de polvo, gas y vapor, se formó nuestro Sistema Planetario, a través de varios millones de años, la mayor parte del gas y el polvo fueron concentrándose, por la acción de la fuerza de gravedad, para formar el Sol, mientras que otros grumos de materia formaron los planetas, sus satélites, los cometas y los asteroides que hoy conocemos.

1.3 El Sol, su estructura y actividad solar.

1.3.1 Descripción general del Sol.

“El Sol es una estrella bastante común, es una estrella enana de color amarillo, miembro de uno de los más numerosos tipos de estrellas las del tipo espectral G2¹” (Blanca Mendoza, 1993; p. 10).

Julieta Fierro (1988) nos describe en su texto, La familia del Sol, a nuestra estrella de la siguiente forma.

El Sol es el componente más notable del Sistema Solar, provee la mayor parte de energía a nuestro Sistema Planetario además le pertenece la mayor parte de la masa del Sistema, el Sol está en el centro de la masa del Sistema Planetario y todos los cuerpos gravitan a su alrededor. El Sol, a su vez, se mueve junto con su Sistema, en relación con las estrellas de su vecindad, y también alrededor del centro de la Galaxia, a una velocidad de 250 km/seg.

Durante los últimos 4,600 millones de años la Tierra ha estado ligada gravitacionalmente al Sol y este la ha bañado de energía, en la actualidad la vida se mantiene gracias a que nuestro Planeta puede capturar y almacenar la energía proveniente de la estrella. La fuente de energía del Sol proviene de la fusión nuclear que se lleva a cabo en su interior, en la que se está transformando continuamente dos átomos de deuterio (que es un isótopo del hidrógeno) en uno de helio, utilizando como catalizador al carbono. Debido a que la masa de dos átomos de deuterio es ligeramente mayor a la masa del átomo de helio, el exceso de masa se transforma en energía, de acuerdo con la relación $E = mc^2$. La masa transformada en energía por reacción, es tan sólo de 0.007 veces la masa original. Cada año, el Sol transforma un diezmillonésimo de millonésimo de su masa en energía.

Debido a que la masa del Sol es casi constante, y por la regularidad de las reacciones nucleares, el Sol se ha mantenido a la misma temperatura desde hace 4,500 millones de años y lo seguirá haciendo durante otro periodo similar, esto ha permitido, en parte, que se haya podido desarrollar la vida inteligente en la Tierra.

El Sol es una esfera de gases muy calientes que está rotando y que posee un campo magnético entre 1 y 2 gauss, se está evaporando; como resultado de la evaporación se produce el viento solar, que es un conjunto de partículas cargadas (núcleos de hidrógeno, helio y electrones) que arrastran consigo líneas de campo magnético.

¹ Las estrellas son clasificadas por una letra y un número que describen la naturaleza de sus líneas espectrales y la temperatura de su superficie. Los tipos hoy en día clasificados, son: O, B, A, F, G, K y M, las estrellas de más alta temperatura son las de tipo O y las de más baja temperatura son las de tipo M, el número solo indica una subdivisión entre estrellas de una misma clase.

4 Capítulo 1. El Sol, actividad solar y emisiones solares

Las características físicas del Sol se listan en la siguiente ficha técnica:

Propiedad Física	Magnitud
Edad	$\approx 4.5 \times 10^9$ Años
Masa	$M_{\oplus} = 1.99 \times 10^{30}$ Kg.
Radio	$R_{\oplus} = 6.96 \times 10^8$ m.
Densidad Media	1.4×10^3 kg m ⁻³
Distancia Media Tierra-Sol	1 AU = 1.50×10^{11} m = 215 R_{\oplus}
Gravedad en la superficie	$G_{\oplus} = 274$ m s ⁻²
Luminosidad	$L_{\oplus} \approx 3.86 \times 10^{26}$ W
Periodo de Rotación ecuatorial	26 Días
Temperatura Efectiva	5785 K

Tabla 1.1: Ficha técnica del Sol (Eric Priest, 1982, pp. 3,4)

“La espectroscopia ha permitido, desde el siglo pasado, la identificación de elementos químicos que constituyen el Sol. La mayor parte del Sol es hidrógeno, aproximadamente el 92% de sus átomos son de este elemento, alrededor de un 7.9% es helio, los demás elementos constituyen el 0.1 %” (Silvia Bravo, 1987; p. 64).

1.3.2 Estructura del Sol, producción y transporte de energía.

El Sol es una esfera de gas caliente, no es una esfera homogénea, tiene una estructura diferenciada en capas (ver figura 1.3), las cuales tienen propiedades y características particulares, que a continuación se describen.

El Sol se divide en dos secciones principales la región interna, en la cual se encuentran contenidos el núcleo, la zona radiativa y la zona convectiva, a la otra región se le denomina atmósfera solar y contiene a la fotosfera, a la cromosfera y a la corona (ver figura 1.3).

A continuación se describen cada una de las regiones con sus respectivas capas:

1.3.2.1 Región Interna.

La región interna (ver figura 1.2) del Sol no se puede observar directamente, solo se puede deducir por consideraciones teóricas a partir de sus características superficiales.

1.3.2.1.1 El Núcleo.

El núcleo del Sol es extremadamente denso, es aquí donde tiene lugar la producción de energía de la estrella de nuestro Sistema Planetario.

Robert W. Noyes nos describe la producción de energía al interior del núcleo del Sol de la siguiente forma.

El Sol genera su energía por medio de la fusión del núcleo del hidrógeno, o protón, con el núcleo de helio. En el proceso, una diminuta fracción de materia se convierte en energía pura y es liberada. Cada segundo, el Sol procesa cerca de 700 millones de toneladas de hidrógeno, convirtiéndolas en helio, y durante esa *quemazón* nuclear, casi 5 millones de toneladas de materia se transforman en energía pura. La explicación para este proceso es la siguiente: El núcleo de un átomo de helio está constituido por 2 neutrones y dos protones y tiene la designación química de ${}^4\text{He}$, el número 4 es el número total de protones y neutrones. Sin embargo la suma de las masas de 2 protones y 2 neutrones, al medirse como partículas separadas, es ligeramente mayor que la masa de un núcleo de helio. La discrepancia es leve, solamente 7 partes en mil, este número de partes se convierte en energía, liberada cuando se constituye el núcleo de helio. Por cada núcleo de helio formado, una energía igual al 0.7 % de $M_{\text{He}}c^2$, donde M_{He} es la masa del núcleo de helio, se adecua para proveer de energía al *horno solar*. Como la luminosidad del Sol es aproximadamente 4×10^{33} ergs/segundo², y aplicando la ecuación de Einstein ($E=mc^2$), tenemos que la masa convertida en energía cada segundo es:

$$E = (4 \times 10^{33} \text{ ergs/segundo}) / (3 \times 10^8 \text{ metros/segundo})^2$$

$$E = 4.4 \times 10^{12} \text{ gramos/segundo}$$

Es decir cerca de 5 toneladas, lo cual nos lleva a decir que el Sol a lo largo de su vida y hasta la fecha ha perdido casi 5.5×10^{23} toneladas, o bien una fracción de $1/4000$ de su masa, convirtiéndola en energía.

Adentrándonos más en la fusión nuclear, en el texto, El Sol nuestra estrella, encontramos una descripción muy puntual de cómo se lleva a cabo este proceso. “Cada núcleo de helio se ensambla paso a paso a partir de sus nucleones constitutivos (ver figura 1.1). Primeramente, dos protones se fusionan para formar un núcleo intermedio de hidrógeno de *peso doble*, o deuterio, denotado como ${}^2\text{H}$ el cual consta de un protón y un neutrón. (En el proceso de convertir uno de los protones en un neutrón su carga electrónica positiva es llevada por un electrón positivo, o positrón; el positrón se encuentra inmediatamente con un electrón ordinario y ambos se convierten en un par de rayos gamma energizados). Como segundo paso, otro protón se fusiona con el núcleo de deuterio (esta vez sin soltar un positrón pero liberando energía en forma de rayos gamma) para formar un núcleo denominado ${}^3\text{He}$, o helio *ligero*, que consta de dos protones y un neutrón. Finalmente, dos núcleos de ${}^3\text{He}$, formado cada uno de esta manera, se fusiona para formar un núcleo de ${}^4\text{He}$ (helio ordinario) que consta de dos protones y dos neutrones, los dos protones restantes escapan para tomar parte en el proceso nuevamente. En cada paso de esta secuencia se libera energía; la totalidad liberada en todos los pasos alcanza el valor de $0.007 M_{\text{He}}c^2$ ya mencionado. Otras formas de fusión de helio a partir de protones ocurren en el núcleo solar; solamente se ha descrito aquí la forma más directa” (Robert W. Noyes, 1985, pp. 88-90).

² El erg es la unidad de energía del sistema csg (centímetro-segundo-gramo), y es igual a $1 \text{ g} \cdot \text{cm}^2 / \text{s}^2$, y se define como Trabajo realizado por una fuerza de 1 dyn, cuando su punto de aplicación se desplaza 1 cm en su misma dirección. 1 erg = 1 dyn x 1 cm.

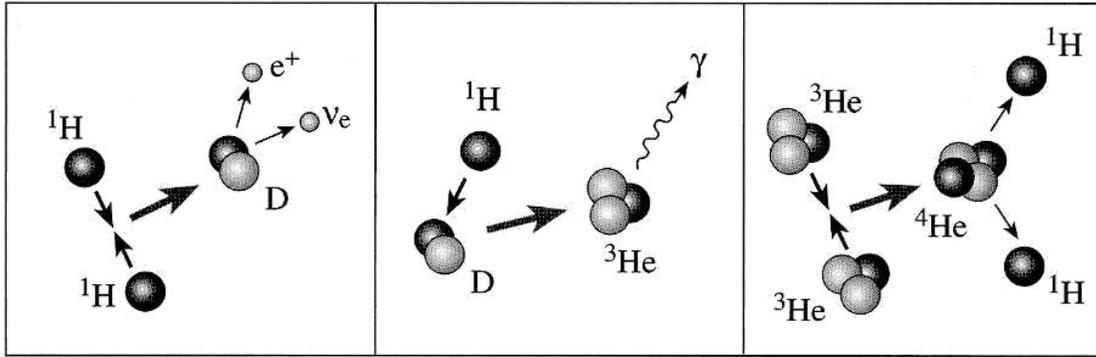


Figura 1.1: Cadena protón-protón

1.3.2.1.2 La zona radiactiva.

“Esta zona es la que envuelve al núcleo del Sol y comprende aproximadamente de ${}^{2/10} R_{\oplus}$ hasta, posiblemente, ${}^{6/10}$ u ${}^{8/10} R_{\oplus}$. Los rayos gamma³ que se producen en las reacciones nucleares en el Sol degeneran muy pronto en rayos X⁴ que se dirigen hacia la superficie a través de la zona radiactiva. En esta zona los fotones de radiación⁵ X van sufriendo una gran cantidad de colisiones con iones y electrones y van perdiendo con ellas energía” (Silvia Bravo, 1987, p. 91).

1.3.2.1.3 La zona convectiva.

“En la zona de convección⁶, es el material caliente el que fluye hacia arriba, transportando la energía hacia la superficie donde se producen los fotones que finalmente son emitidos hacia el espacio en forma de luz y calor” (Silvia Bravo, pp. 91,92). Esto es, en esta zona la luz (los fotones) es absorbida por átomos en estado gaseoso, y se producen fuertes corrientes de convección (similares, aunque guardando las proporciones, a las que se generan en los océanos y la atmósfera terrestre), que se encargan de transportar la energía que proviene de la zona de radiación hacia la fotosfera, como la temperatura del gas que ha absorbido energía aumenta, el gas se expande, haciéndose por ello menos denso, debido a ello estas *bolsas* de gas ascienden hacia la parte superior de la capa convectiva, una vez que llegan ahí expulsan la energía acumulada, pasando a ser de nuevo gas menos caliente y menos denso, por esta característica el gas vuelve a descender a través de la zona convectiva hacia su parte inferior, este ciclo dura varios meses.

³ La radiación gamma es un tipo de radiación electromagnética producida por elementos radioactivos o procesos subatómicos como la aniquilación de un par positrón-electrón.

⁴ Los rayos X son una radiación electromagnética, invisible, capaz de atravesar cuerpos opacos, se producen a nivel extranuclear, por fenómenos de frenado electrónico.

⁵ La radiación consiste en la propagación de energía en forma de ondas electromagnéticas o partículas subatómicas a través del vacío o de un medio material

⁶ La convección es el transporte de energía por medio de las corrientes ascendente y descendente en un fluido.

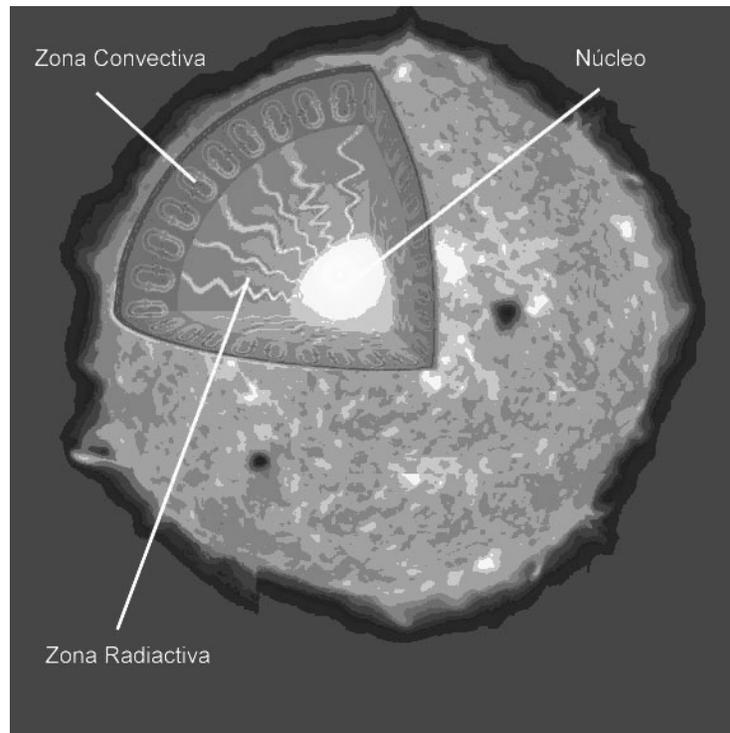


Figura 1.2: Región interna del Sol

1.3.2.2 La Atmósfera Solar.

A diferencia de la región interna, la región externa del Sol, o bien, atmósfera solar, si se ha podido observar, por lo cual se tiene un panorama amplio de la mayor parte de los fenómenos solares que ocurren en esta región, aunque algunos de ellos son sumamente complejos y no se han podido modelar.

1.3.2.2.1 La Fotosfera.

La mayor parte de la energía que recibe la Tierra proviene de la fotosfera, que emite un continuo de radiación electromagnética, casi toda en el espectro visible. La parte más baja de la fotosfera está compuesta por material parcialmente ionizado⁷, en su mayor parte hidrógeno, en la parte alta de esta capa se tiene hidrógeno principalmente neutro. Esta capa es muy delgada y contiene aproximadamente un quinto de billonésima parte de la masa del Sol. En la fotosfera la temperatura disminuye de abajo hacia arriba de 8,500 K, en la parte inferior, hasta 4,500 K en la parte superior.

Es en la fotosfera donde se pueden observar algunos fenómenos que nos indican el constante cambio que hay al interior del Sol, me refiero a los gránulos, supergránulos y manchas solares, los primeros tienen un tamaño promedio de 2,000 km y son de vida muy corta, alrededor de 10 minutos, los segundos de aproximadamente 30,000 km de diámetro y cuya vida es de 1 día aproximadamente, por último, las manchas solares, son la característica más

⁷ Un ion es un átomo o molécula cargada positiva o negativamente, esto es debido que durante alguna reacción perdió o ganó electrones.

8 Capítulo 1. El Sol, actividad solar y emisiones solares

notable de esta capa del Sol, y se describen como enormes regiones oscuras con tamaños entre 1,000 y 100,000 kilómetros que rotan con el Sol y cuyo número aumenta y disminuye siguiendo un ciclo de aproximadamente 11 años. En la fotosfera también aparecen las llamadas fáculas, que son regiones más brillantes y más calientes que el resto de la fotosfera y que suelen estar asociadas a las manchas. El exceso de temperatura de una fácula es a lo más de 250 K.

1.3.2.2.2 La Cromosfera.

La descripción que a continuación hago de esta capa del Sol esta basada en la realizada por Silvia Bravo en su texto, Encuentro con una estrella.

Cuando sucede un eclipse total de Sol, la Luna bloquea la intensa luz de la estrella. En esos momentos es posible observar un anillo de color rojo magenta alrededor del disco cubierto, esta coloración fue la causa del nombre que tiene, ya que significa esfera de color, la cromosfera tiene un grosor de entre 1,000 y 8,000 kilómetros; hacia la parte baja de esta capa se tiene una temperatura aproximada de unos 4,000 K y sus primeros 3,000 kilómetros están compuestos de átomos neutros de hidrógeno y se tiene una densidad del orden de un billón de átomos por centímetro cúbico. Después de los 3,000 kilómetros de grosor la temperatura empieza a subir súbitamente hasta alcanzar un valor de 1 millón de Kelvin's en los 8,000 kilómetros, a esta altura la densidad ha bajado hasta unos 1 000 millones de átomos por centímetro cúbico y toda la materia se encuentra ionizada. A la parte más alta de la cromosfera se le conoce como región de transición y es ahí donde empieza la corona. Cabe mencionar que el gas que forma la cromosfera es prácticamente transparente a la luz fotosférica, no es posible observarlo en luz blanca, salvo en los momentos de un eclipse total de Sol, debido a esto, para obtener detalles de la cromosfera se utiliza el espectroheliógrafo, dicho instrumento permite ver al Sol en longitudes de onda donde la cromosfera emite intensamente, como por ejemplo, en una de las líneas del hidrógeno (la llamada $H\alpha$) de $6\,563\text{ \AA}^8$, en la parte roja del espectro. Un aspecto muy importante es que toda la estructura de esta capa está determinada por el campo magnético del Sol, esto se debe a que la mayor parte de la materia en ella está ionizado y la presión del gas es muy baja comparada con la presión magnética, por lo que las líneas del campo controlan y ordenan los movimientos del material.

1.3.2.2.3 La Corona.

La corona es la capa más extensa del Sol, nace desde la parte en que termina la cromosfera y se extiende a lo largo del sistema planetario, es decir los planetas de nuestro sistema se encuentran inmersos en la corona, "nombrada así porque en un eclipse total de Sol, esta se observa como una tenue luz blanca aperlada, coronando el disco oscurecido" (Silvia Bravo, 1987, p.71).

En la corona solar prevalecen condiciones de muy altas temperaturas y baja densidad, la materia en esta capa se encuentra ionizado, además hay una gran

⁸ Un amstrong \AA , equivale a 10^{-10} metros, se utiliza principalmente para definir longitudes de onda muy pequeñas.

cantidad de electrones libres que se mueven a gran velocidad, estos electrones dispersan la luz fotosférica, tal dispersión es la que produce el brillo blanquecino de esta capa. El plasma coronal aumenta tanto su temperatura que a una cierta altura ya no puede ser retenido, lo cual da origen al viento solar.

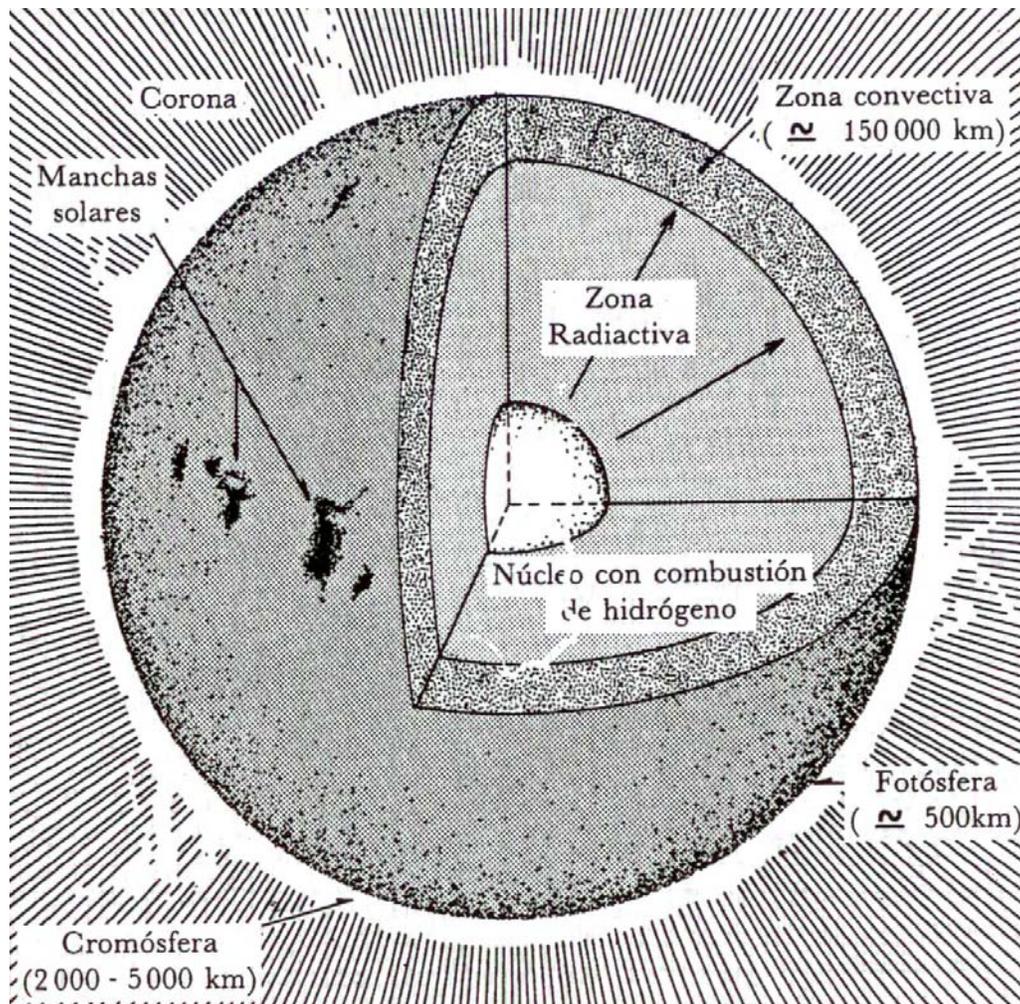


Figura 1.3: Las capas del Sol, corte esquemático. (Dibujo de Alberto García)

1.3.3 Actividad solar.

Una descripción muy larga podría darse para definir lo que es la actividad solar, de lo leído en varias fuentes bibliográficas definiré a la actividad solar, únicamente, como todas aquellas manifestaciones que varían con el tiempo y que tiene cabida en la atmósfera solar.

El indicio principal de actividad solar son las manchas solares, las cuales se describirán más adelante.

1.3.3.1 La rotación diferencial del Sol.

Al respecto de la Tierra, sabemos que cumple cíclicamente con dos movimientos, el de translación (giro alrededor del Sol en una órbita elíptica) y el de rotación (giro

10 Capítulo 1. El Sol, actividad solar y emisiones solares

sobre su propio eje, el cual atraviesa los polos), este último tiene características constantes durante todo su ciclo, es decir se realiza en una sola dirección y toda la masa terrestre se mueve a la misma velocidad, hablando de sus diferentes puntos (ecuador, polos, etc), es decir, toda la masa terrestre se mueve de la misma manera. En el Sol no ocurre así, la rotación del Sol es una rotación diferencial, las regiones ecuatoriales giran más rápido que los polos, completando una vuelta en menos días que las regiones polares.

1.3.3.2 El campo magnético del Sol.

Debido a la rotación diferencial, el campo magnético del Sol (ver figura 1.4) no tiene una *forma* constante, oscila entre un dipolo magnético y un toroide magnético, el cambio del campo magnético se produce debido al movimiento macroscópico de plasma en la zona convectiva, el cambio del campo magnético esta asociado al ciclo de manchas solares. El campo magnético cambia, de un dipolo magnético cuyas líneas de campo se encuentran en una dirección, después de once años, a un toroide magnético, tras otros once años, el toroide cambia de nueva cuenta a un dipolo, pero cuyas líneas de campo se encuentran en dirección opuesta al dipolo del ciclo anterior. Es decir, se observan dos ciclos, uno de once años en el cual existe un cambio en la forma del campo magnético, y uno de 22 años en el cual el campo magnético vuelve a tomar la forma de un dipolo.

Tanto la rotación diferencial como el cambio en la forma de campo magnético producen una serie de manifestaciones de actividad solar, las más importantes se describen a continuación.

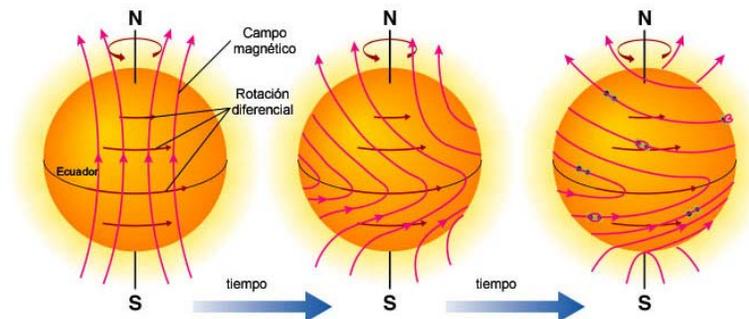


Figura 1.4: El campo magnético solar.

1.3.3.3 Manchas solares.

Las manchas solares son la manifestación visible de que el campo magnético es muy dinámico, y que da origen a una amplia gama de manifestaciones de actividad solar. Las manchas aparecen solas y con configuraciones magnéticas especiales, que van desde las unipolares, bipolares hasta otras mucho más complejas. En las manchas solares se distinguen varias regiones. La umbra es el centro de la mancha, en esta región el campo magnético es mayor, la rodea otra región denominada penumbra y se compone de filamentos oscuros. El aspecto

oscuro de las manchas solares es debido a que su temperatura es menor a la del resto de la fotosfera, es decir, la fotosfera irradia tanta luminosidad, que hace ver a las manchas solares como regiones oscuras, por lo tanto podemos entender que las manchas solares se encuentran en la fotosfera, una mancha solar es una depresión en la fotosfera de unos cuantos cientos de kilómetros de profundidad.

1.3.3.4 Protuberancias.

Son fenómenos que se observan en el limbo del Sol como arcos flameantes en la alta cromosfera y en la corona inferior. Son nubes de materia a temperatura inferior y densidad más alta que la de su alrededor.

En longitudes de onda como ultravioleta o rayos X, se ven como manchas oscuras porque la emisión de la materia coronal es mucho mayor.

Los filamentos son del mismo tipo que las protuberancias, y se los distingue por la manera en que se los ve: absorción o emisión.

Las protuberancias se distinguen entre quiescentes y activas.

1.3.3.5 Filamentos.

Son tubos y bucles de materia relativamente concentrada a temperaturas cromosféricas por encima de los grupos de manchas y pueden llegar a más de 100,000 Km de longitud. Se forman a lo largo de la línea neutra que divide la polaridad opuesta entre dos grupos de manchas.

Sobre el fondo brillante del disco solar, absorben la luz, y vistos en el limbo son unas 100 veces más brillantes que el fondo de la corona y se ven en emisión, entonces se conocen como protuberancias.

Puede crecer unos 10 000 Km a lo largo de una rotación solar y llegar a estirarse hasta 1,000,000 Km.

Después de desaparecer las manchas los filamentos se corren hacia los polos y sobreviven más tiempo que las manchas.

1.3.3.6 Playas.

Son áreas de brillo más intenso visibles en varias líneas monocromáticas y se destacan sobre el fondo de la cromosfera. Su posición coincide aproximadamente con la de las fáculas fotosféricas, por eso se las llama fáculas cromosféricas.

Se cree que su brillo se debe al incremento del flujo de energía en la atmósfera solar que es transportada por la acción del campo magnético concentrado. También se supone que la fuente principal de calentamiento de la atmósfera solar es la deposición de energía por medios mecánicos, por eso el proceso que se encarga de aumentar la temperatura de las playas es el entramado cromosférico.

Los campos magnéticos en las playas son de 100 a 200 G⁹, y eso permite el incremento de deposición de energía en ellas y en la corona encima de ellas, al contrario de lo que ocurre con las umbras de las manchas.

⁹ 1 G = 1 Gauss, es la unidad de campo magnético,

1.3.3.7 Fulguraciones.

Una fulguración es una de las manifestaciones de actividad solar, y se define como una súbita liberación de energía almacenada en los campos magnéticos asociados con una región activa. Esta energía se almacena en los retorcimientos y reforzamientos del campo magnético de fuertes regiones activas, el agente que produce el retorcimiento de los campos es el movimiento caótico en la zona de convección, debido a ello la fuente de energía resulta siempre del interior del Sol.

La energía liberada en una fulguración, se debe a la cantidad de energía acumulada en las regiones activas en donde se originan, una vez acumulada la energía existe una liberación súbita de ella, es decir, la energía se acumula y se concentra en una región definida, posteriormente algún suceso hace que la configuración magnética se desestabilice, de pronto, un canal se abre para la liberación de energía que se lleva a cabo a gran velocidad, en una diminuta fracción de segundo, produciéndose una fulguración.

La energía liberada, en forma de electrones de altas velocidades, toca la cromosfera simultáneamente en un rápido incremento de $H\alpha$ y de otros tipos de emisiones cromosféricas, si la fulguración es suficientemente intensa, la explosión de electrones puede penetrar las densas capas de la superficie produciendo un rayo de luz visible. Existen varios escenarios posibles para una fulguración, pero lo explicado anteriormente es lo que a grandes rasgos sucede, a continuación se explica uno de los posibles escenarios de una fulguración describiendo la forma en que la energía es liberada.

Una vez que la energía es almacenada en una región activa y bajo la influencia de algún otro fenómeno que afecte a tal acumulación de energía, esta es desprendida mediante un arco o loop magnético dibujado a través de plasma a alta temperatura, en el proceso la energía se libera formando espirales de energía lo cual da lugar a liberación de rayos gamma, ultravioleta y rayos X, además de partículas aceleradas (protones y electrones) y microondas de radio, este escenario se representa en la figura 1.5.

Una manera simple de definir a una fulguración es la siguiente.

Una fulguración es una luz intensa en las longitudes de onda visible, rayos X, entre otras, se debe a la aniquilación de campos magnéticos la cual provoca una súbita liberación de energía que se ve en varias longitudes de onda y en aceleración de partículas.

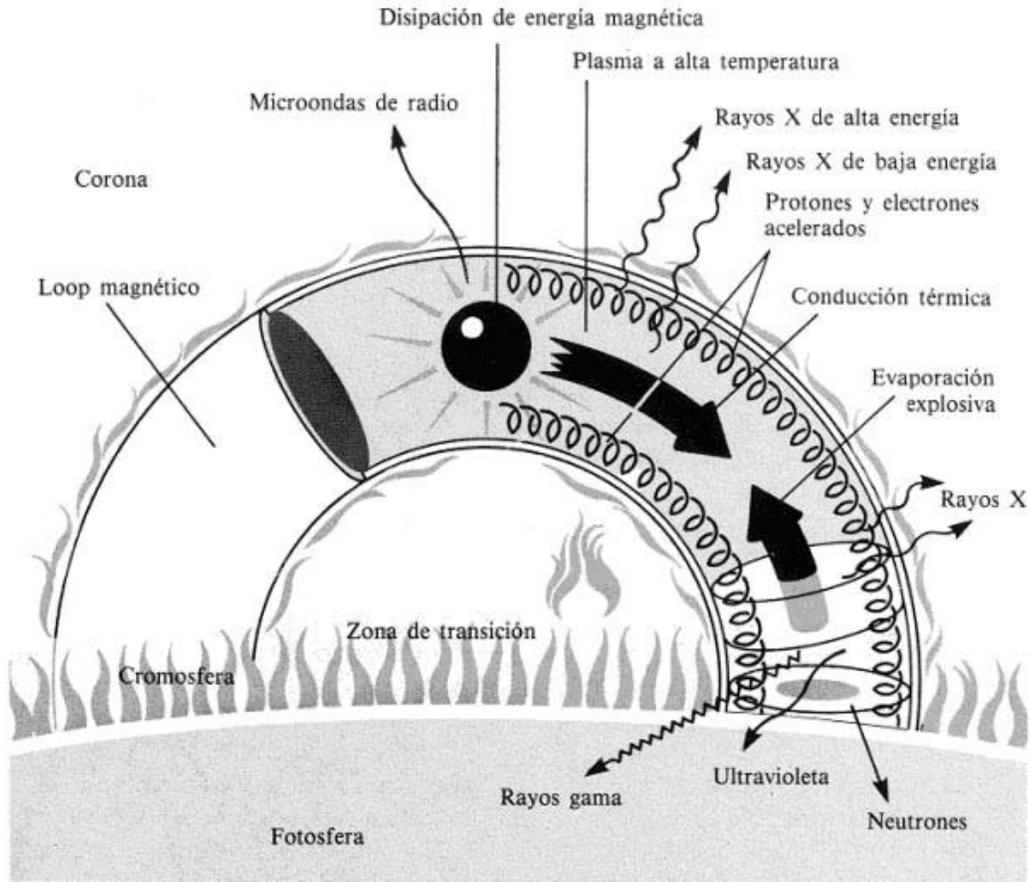


Figura 1.5: Posible escenario para una fulguración.

1.3.3.8 Eyecciones de masa coronal.

Las eyecciones de masa coronal son los fenómenos eruptivos más interesantes, y aunque no se sabe a ciencia cierta que los causa, con la ayuda de los cronógrafos se ha podido estudiar a detalle las estructuras de plasma y campo magnético que rodean al Sol. En general la evolución de esas estructuras es muy lenta, pero, en ocasiones se observa como una parte de la corona solar baja se desprende y es lanzada hacia el medio interplanetario, este fenómeno es lo que se conoce como una eyección de masa coronal (EMC) que podría definirse como una reconfiguración del campo magnético coronal de gran escala. Este tipo de fenómenos es muy frecuente en el Sol, en máximos del ciclo solar se llegan a observar varias veces al día, la cantidad de masa eyectada varía mucho, pero puede llegar hasta unos 10^{16} g, con velocidades que van desde unos cuantos metros por segundo hasta varios cientos de metros por segundo.

Por sus efectos en la Tierra y su entorno, las EMC son los fenómenos eruptivos solares que producen más perturbaciones observadas en el entorno magnético terrestre, en un principio se pensaba que estas perturbaciones eran debidas a las fulguraciones solares.

14 Capítulo 1. El Sol, actividad solar y emisiones solares

Hoy en día se sabe que ambos fenómenos (EMC y fulguraciones solares) están estrechamente relacionados y cuando suceden ambos casi al mismo tiempo provocan la emisión de partículas cargadas y neutrones hacia el medio interplanetario, incluyendo la Tierra, tales emisiones pueden perturbar las comunicaciones terrestres, en especial, las satelitales.

Ya se describió en la sección del campo magnético solar que existen ciclos bien definidos en la actividad solar, cada once años la actividad solar oscila entre un mínimo y un máximo, en el mínimo de actividad solar se espera que haya pocas manchas solares así como pocas fulguraciones y eyecciones de masa coronal, en un máximo se espera que haya muchas manchas solares, así como mayor cantidad de fulguraciones y eyecciones de masa coronal, así como otras manifestaciones de la actividad solar.

1.3.3.9 Viento solar.

Aunque no se saben las causas por las cuales la corona es tan caliente, la consecuencia inmediata se ha podido observar, un gas tan caliente no puede quedar confinado en un espacio definido, es por ello que se extiende muy lejos de donde inicia la capa coronal debido a que la fuerza gravitacional en el Sol es incapaz de retenerlo. Podemos decir que el viento solar es un continuo fluir de la corona solar. Como ya se mencionó la corona esta compuesta por material ionizado, eléctricamente este material es neutro, no así su composición ya que esta conformado por iones que tienen carga eléctrica positiva y electrones con carga negativa, es decir, lo que compone al viento solar son partículas cargadas las cuales son susceptibles de ser afectadas o afectar a campos magnéticos y eléctricos. Uno de los efectos más importantes del viento solar debido a su alta conductividad eléctrica es que de alguna manera transporta el campo magnético del Sol hacia el medio interplanetario, estirando las líneas de campo, es por ello que el campo magnético del Sol no se cierra cerca de él.

Así mismo las partículas cargadas que conforman al viento solar interactúan con los campos magnéticos de los planetas, por ejemplo con el de la Tierra, es por ello que el viento solar es una fuente de emisiones solares que se han podido detectar en la Tierra.

Las condiciones de temperatura, densidad y presión juegan un papel sumamente importante en la producción de energía que se da desde el interior del Sol, y en la forma en que esta es transportada a lo largo de las capas que lo conforman, la última manifestación de actividad solar descrita, quizá tenga su principal origen en la altísima temperatura que reina en la capa coronal, no solo el viento solar es afectado por las condiciones de temperatura, también mencionamos que la causa de la coloración oscura de las manchas solares es debido a que en ellas la temperatura es menor al resto de la fotosfera; por considerarlo de interés a continuación les presento una gráfica (figura 1.6) en la cual se observa la variación de la temperatura a lo largo de las capas atmosféricas de nuestro astro Rey.

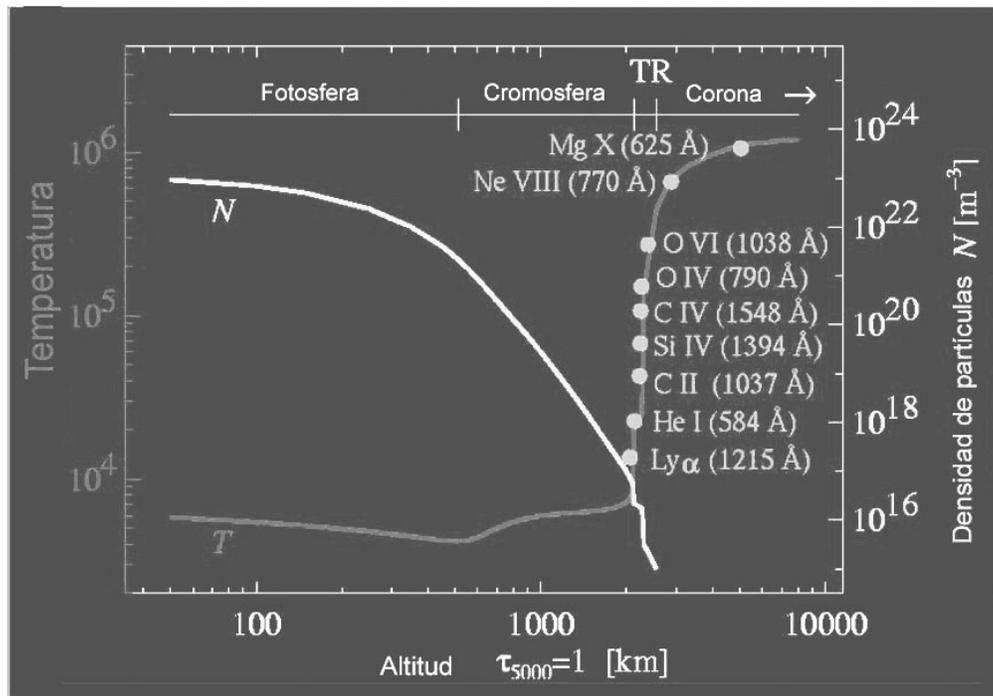


Figura 1.6: Variación de densidad y temperatura en la atmósfera solar.

1.3.3.10 Emisiones solares.

Cuando se habla de energía emitida por el Sol se hace alusión a ondas electromagnéticas y partículas cargadas. Las ondas electromagnéticas que se emiten desde el Sol cubren un amplia gama del espectro electromagnético, que va desde las ondas de radio, pasando por las microondas, el infrarrojo, el visible, los ultravioleta, hasta los rayos X y rayos γ , pasando por longitudes de onda de 10,000 (para las ondas de radio) hasta $1/10^8$ (para los rayos γ). Cabe mencionar que no todas las emisiones solares llegan a la Tierra, algunas de ellas son *bloqueadas* por nuestra atmósfera en la Tierra.

Metafóricamente hablando, "el Sol es una gran máquina la cual produce gran cantidad de energía mediante la liberación súbita y aceleración de partículas, de la actividad que presenta el Sol es en las fulguraciones y en las eyecciones de masa coronal en donde se produce este tipo de energía, la mayoría de las emisiones solares que se generan por fulguraciones solo se pueden detectar a través de naves espaciales, pero en los intervalos centimétrico y milimétrico se pueden observar desde la Tierra, el intervalo milimétrico requiere de grandes alturas. Las pocas observaciones de fulguraciones solares en el intervalo milimétrico indican una correlación muy cercana con emisiones de muy alta energía como rayos X duros y rayos gama.

Las partículas aceleradas pueden tener interacciones nucleares con el gas de la atmósfera solar y producir neutrones y rayos gama. Algunos neutrones no pueden escapar de la atmósfera solar y son entonces detectados en líneas de emisión de rayos gama. Los que escapan pueden ser detectados en el espacio o incluso en la Tierra. Las mediciones de neutrones solares en la superficie terrestre son un indicador de la existencia de iones acelerados hasta energías de algunos GeV

(10^9 eV)¹⁰ en la atmósfera solar. La detección eficiente de estos neutrones nos da información importante para conocer detalles tanto del proceso de aceleración como de las densidades y composición química de la atmósfera solar. Hasta hoy en día solo ha sido posible observar unos cuantos eventos de neutrones, debido principalmente a que no existen en operación suficiente número de detectores apropiados. Los detectores deben instalarse en sitios elevados y, de preferencia, en latitudes ecuatoriales, donde el tiempo de exposición al Sol es mayor y más uniforme a lo largo del año. Los neutrones solares y las emisiones en radio proporcionan dos ventanas de oportunidad para realizar observaciones de fenómenos importantes y aun no muy bien conocidos. Nuestro país cuenta con al menos un sitio adecuado para hacer estas observaciones: El Telescopio de Neutrones Solares, en Sierra Negra Puebla.” (Información extraída del artículo electrónico: Observatorio Solar Mexicano de Gran Altura, <http://www.geofisica.unam.mx/isyp/osga.html>).

1.3.3.11 Detección de partículas cargadas.

El término *rayos cósmicos* se puede definir como: partículas subatómicas que proceden del espacio exterior y que tienen una energía elevada debido a su gran velocidad, a menudo se suele utilizar el término de manera indistinta, para los propósitos de este trabajo, haremos la distinción solamente entre dos tipos de ellos, los rayos cósmicos galácticos y los rayos cósmicos solares, las principales diferencias entre ambos son su composición, energía y obviamente su origen, independientemente de que tipo de rayos cósmicos tratemos, algunos de ellos pueden llegar a tener interacciones nucleares con la atmósfera terrestre y así producir partículas cargadas que son susceptibles de ser detectadas en la superficie de la Tierra, alguna de ellas se pueden detectar a grandes alturas y otras se pueden detectar incluso a nivel del mar.

Para detectar este tipo de partículas cargadas se han empleado varios tipos de instrumentos, los cuales se describen a continuación.

Detector de agua.

Los detectores de agua (ver figura 1.7) consisten en una caja de paredes de material de polietileno reflectante totalmente cerradas y selladas que no permiten el ingreso de luz y en su interior almacenan una bolsa con agua. Cuando las partículas cargadas atraviesan las paredes de la caja y pasan por el agua producen una radiación que genera unos flashes de una tenue luz azul, como una onda de choque. Esta luz rebota en las paredes reflectivas y, finalmente, es detectada por un fotomultiplicador (un detector de luz muy sensible) que se halla en la parte superior de la caja y, luego, la señal es retransmitida y capturada por algún equipo electrónico o de cómputo que lleva el registro de la detección.

¹⁰ Un electron volt (eV) es la energía que un electrón adquiere al pasar a través de una diferencia de potencial de exactamente un volt, $1 \text{ eV} = 6 \times 10^{-19}$ joules, $1 \text{ J} = 1 \text{ N} \cdot \text{m}$, un J Se define como el trabajo realizado por la fuerza de 1 newton en un desplazamiento de 1 metro.

Detector fototubo multiplicador.

El detector fototubo multiplicador (ver figura 1.8) consiste en una matriz bidimensional confeccionada con un plástico centellador detector de partículas cargadas. Cuando un electrón atraviesa uno de estos plásticos una pequeña parte de su energía se emplea en excitar algunas de sus moléculas las cuales inmediatamente se desexcitan emitiendo un breve flash de fotones. Estos fotones son detectados por un sensible dispositivo fotoeléctrico (fotomultiplicador), el cual produce una señal eléctrica de corta duración. Si se emplea un dispositivo lo suficientemente rápido, se puede medir además el instante de tiempo en que el electrón atravesó el plástico con una precisión del orden de nanosegundos (10^{-9} s). Se trata de detectores capaces de proporcionar información útil para deducir la energía que contenía el rayo cósmico primario (partícula cargada original, partícula acelerada originada en el espacio). Por otro lado, la comparación de los tiempos de arribo de las diferentes partículas a los distintos contadores permite medir la inclinación de llegada, de este modo, la dirección de cada uno de los rayos cósmicos primarios que han sido detectados a su llegada. Esta clase de instrumentos detectan rayos cósmicos provenientes de cualquier lugar del cielo, permitiendo además, al analizar los datos, determinar el ángulo con el que cada rayos cósmico primario (partícula cargada originada en el espacio y que interactuó con la atmósfera terrestre) llegó a la Tierra. Con estos detectores es factible conseguir una resolución angular típica del orden de 0.5° .

Detector centellante.

El detector centellante (ver figura 1.9) está construido de un plástico especial llamado centellador. Al pasar las partículas cargadas por el centellador excitan los átomos del plástico otorgándoles una cierta energía lo que provoca que las partículas cargadas se retrasen un poco. Luego los átomos excitados pierden esa energía emitiendo algunos fotones de luz. Enseguida, esa luz es detectada por un sensible fotomultiplicador, posteriormente se puede llevar el registro de la detección mediante un complejo electrónico y/o computacional.

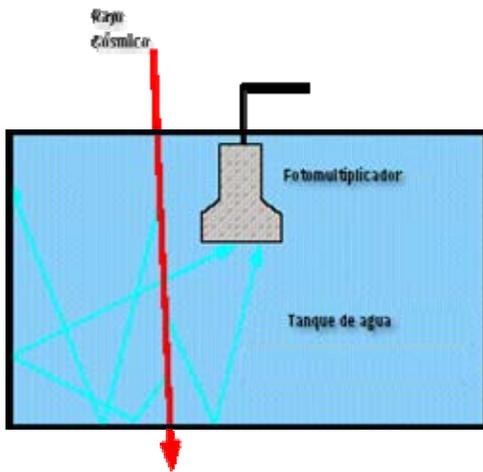


Figura 1.7: Detector de agua.



Figura 1.8: Detector Fototubo multiplicador.



Figura 1.9: Detector Centellante.

De estos instrumentos y varios más se ha valido el hombre para intentar detectar las partículas cargadas que se originan en el espacio y que tienen interacción con nuestra atmósfera terrestre, el objetivo primordial de esto es saber el origen de dichas partículas, sus características y como afectan el medio Terrestre; hablando específicamente de partículas cargadas emitidas por el Sol, el principal objetivo es saber los procesos que ocurren en nuestra estrella y como afectan al resto del sistema planetario, en particular a nuestra Tierra, otro punto importante respecto al estudio de partículas cargadas emitidas por el Sol, es la forma en como se podría aprovechar a futuro la energía proveniente del Sol.

Capítulo 2:

Herramientas de software.



2.1 El software libre.

“El «software libre» es una cuestión de libertad, no de precio. Para comprender este concepto, debemos pensar en la acepción de libre como en «libertad de expresión» y no como en «barra libre de cerveza».

Con software libre nos referimos a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Nos referimos especialmente a cuatro clases de libertad para los usuarios de software:

- Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.
- Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades -el acceso al código fuente es condición indispensable para esto-.
- Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.
- Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad -el acceso al código fuente es condición indispensable para esto-.

Software libre es cualquier programa cuyos usuarios gocen de estas libertades. De modo que deberías ser libre de redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Gozar de esta libertad significa, entre otras cosas, no tener que pedir permiso ni pagar para ello. Asimismo, deberías ser libre para introducir modificaciones y utilizarlas de forma privada, ya sea en tu trabajo o en tu tiempo libre, sin siquiera tener que mencionar su existencia. Si decidieras publicar estos cambios, no deberías estar obligado a notificárselo a ninguna persona ni de ninguna forma en particular. La libertad para utilizar un programa significa que cualquier individuo u organización podrán ejecutarlo desde cualquier sistema informático, con cualquier fin y sin la obligación de comunicárselo subsiguientemente ni al desarrollador ni a ninguna entidad en concreto.

La libertad para redistribuir copias supone incluir las formas binarias o ejecutables del programa y el código fuente tanto de las versiones modificadas como de las originales -la distribución de programas en formato ejecutable es necesaria para su adecuada instalación en sistemas operativos libres-. No pasa nada si no se puede producir una forma ejecutable o binaria -dado que no todos los lenguajes pueden soportarlo-, pero todos debemos tener la libertad para redistribuir tales formas si se encuentra el modo de hacerlo.

Para que las libertades 2 y 4 -la libertad para hacer cambios y para publicar las versiones mejoradas- adquieran significado, debemos disponer del código fuente del programa. Por consiguiente, la accesibilidad del código fuente es una condición necesaria para el software libre.

Para materializar estas libertades, deberán ser irrevocables siempre que no cometamos ningún error; si el desarrollador del software pudiera revocar la licencia sin motivo, ese software dejaría de ser libre.”¹ (Richard Stallman, 2004, Madrid).

¹ Aunque la definición fue sustraída del texto publicado en 2004, la definición originalmente la hizo el autor en 1996.

2.2 El sistema operativo con núcleo Linux.

Un Sistema Operativo es un conjunto de programas que controla la ejecución de programas de aplicación y actúa como una interfaz entre el usuario y el hardware de una computadora, esto es, un Sistema Operativo explota y administra los recursos de hardware de la computadora con el objeto de proporcionar un conjunto de servicios a los usuarios del sistema.

El núcleo de Linux fue creado por Linus Torvalds, un joven universitario que asistía la Universidad de Helsinki, en Finlandia, basándose en el código del sistema operativo *minix* logró desarrollar un pequeño clon de UNIX.

Los primeros sistemas Linux se originaron en 1992, al combinar utilidades de sistema y bibliotecas del proyecto GNU con el núcleo Linux, creado por Torvalds, con la colaboración de cientos de co-desarrolladores y que se distribuye bajo la Licencia Pública General de GNU. Desde finales del 2000 se ha incrementado el apoyo y respaldo de parte de fabricantes de hardware como IBM , Sun Microsystems , Hewlett-Packard, Novell, entre otros.

La marca registrada *Linux* (Número de serie: 1916230) pertenece a Linus Torvalds y se define como "un sistema operativo para computadoras que facilita su uso y operación", además es de tipo software libre, dado que cumple con las 4 reglas mencionadas en la sección 2.1.

Actualmente existen varias *formas* del sistema operativo con núcleo Linux, dichas formas incluyen una enorme variedad de aplicaciones, como: entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etc. A estas formas se les llama distribuciones, hoy en día existen muchas distribuciones del sistema, una de ellas y que hemos elegido como plataforma de trabajo, es Fedora, dicha distribución nació a partir de Red Hat, que en los inicios de Linux fue una de las distribuciones más fuertes; otras de las distribuciones más populares en México son: Debian, Ubuntu, OpenUSE, Mandriva, entre otras.

A este punto cabe hacer la aclaración, que Linux corresponde estrictamente al núcleo aunque en la actualidad se utiliza el nombre para nombrar al núcleo y al sistema operativo de manera indistinta. Las herramientas que suelen incluirse en las distribuciones de este sistema operativo se obtienen de diversas fuentes, como el GNU² y el BSD³ o el KDE⁴. Debido a que las herramientas de software libre que en primera instancia volvieron funcional al núcleo de Linux provienen del proyecto GNU que desde 1983 había liberado software que pudo ser usado en el proyecto de Linux de 1991, Richard Stallman (fundador del proyecto GNU) pide a los usuarios que se refieran a dicho sistema como GNU/Linux. A pesar de esto, la mayoría de los usuarios continúan llamando al sistema simplemente "Linux". La mayoría de los sistemas "Linux" incluyen también herramientas procedentes de BSD y de muchos otros proyectos como Mozilla, Perl, Ruby, Python, PostgreSQL,

² GNU es un acrónimo recursivo que significa GNU No es Unix(*GNU is Not Unix*).

³ BSD son las iniciales de Berkeley Software Distribution (en español, Distribución de Software Berkeley).

⁴ KDE, K Desktop Environment o Entorno de Escritorio K..

MySQL, Xorg, casi todas con licencia GPL⁵ o compatibles con ésta, otro aporte fundamental del proyecto GNU.

Usualmente se utiliza la plataforma XFree86 o la X.Org Server para sostener interfaces gráficas en general.

2.3 Paradigma de Programación Orientada a objetos.

Según el historiador Thomas Kuhn, un paradigma, es un conjunto de teorías estándar y métodos, que juntos representan un medio de organización del conocimiento.

Para contextualizar el enfoque de la programación orientada a objetos debemos mencionar los paradigmas o enfoques más importantes de la programación.

Orientación	Paradigma	Basado en
Orientado a procedimientos.	Paradigma de programación estructurada	Algoritmos
Orientado a objetos	Paradigma de programación orientada a objetos	Clases y objetos
Orientados a lógica	Paradigma de programación lógica	Expresado en cálculo de predicados
Orientado a reglas	Paradigma de programación funcional	Reglas

Tabla 2.1: Orientaciones de los paradigmas de programación

En la programación orientada a objetos las entidades centrales son los objetos, que son tipos de datos que enmarcan con un mismo nombre, sus cualidades (datos) y sus operaciones (métodos).

2.3.1 Características esenciales de la programación orientada a objetos.

La orientación a objetos trata de cumplir con las necesidades de los usuarios finales, así como las propias de los desarrolladores de software. El soporte fundamental es el modelo objeto, los 4 elementos más importantes de este modelo son:

- Abstracción.
- Encapsulamiento
- Modularidad

⁵ GPL: Acrónimo en inglés para, Licencia Pública General

- Jerarquía

Si alguno de estos elementos no se encuentran en el modelo, el modelo, no sugiere ser un modelo objeto.

Abstracción.

Se refiere a la descripción de una entidad del mundo real y utilizar esta descripción en un programa.

Encapsulamiento.

Es el proceso de ocultar todos los secretos de un objeto que no contribuyen a sus características esenciales.

Modularidad.

Esta propiedad permite subdividir una aplicación en partes o pequeños módulos, cada módulo debe ser tan independiente como sea posible de los demás.

Jerarquía.

Esta propiedad permite una ordenación de las abstracciones. Las jerarquías de un sistema complejo son: estructura de clases (es un) y la estructura de objetos (es parte de). La jerarquía de generalización-especialización se conoce como herencia. La herencia la podemos entender como una relación en la que una de las clases comparte la estructura o comportamiento definido en otra u otras clases.

Polimorfismo.

El modelo objeto, no obliga, pero supone como parte esencial de él la propiedad de polimorfismo la cual nos sugiere que podamos referirnos a objetos de clases diferentes mediante el mismo elemento de programa y realizar la misma operación de diferentes formas, según sea el objeto que se referencia en ese momento. Por ejemplo, el polimorfismo implica la posibilidad de tomar un objeto de un tipo "impresora" e indicarle que ejecute imprimir, esta acción se ejecutara de diferente forma, según sea el objeto impresora sobre el que se aplica (impresora de tinta negra, impresora a color, plotter, etc.).

La programación orientada a objetos nos brinda varias ventajas, de las cuales las más importantes se enlistan a continuación:

- **Mantenibilidad:** Facilidad de mantenimiento.
- **Modificabilidad:** Facilidad para modificar los programas.
- **Reusabilidad.** Los objetos se pueden usar numerosas veces y en distintos proyectos.
- **Confiabilidad.** Los programas orientados a objetos suelen ser más fiables ya que se basan en el uso de objetos ya definidos que están ampliamente probados.

Estas ventajas son directas a los programadores. Pero la orientación a objetos no sólo reporta beneficios a los programadores. En las etapas de análisis, previas a la codificación, el utilizar un modelado orientado a objetos reporta grandes beneficios

ya que estas mismas ventajas son aplicables a todas las fases del ciclo de vida de un proyecto software.

Para adentrarnos y entender un poco más acerca del paradigma de programación orientada a objetos es necesario definir algunos conceptos básicos.

- **Clase:** Es una abstracción que define la funcionalidad y los elementos que constituyen a una serie de objetos.
- **Objeto:** Un objeto es una cosa, generalmente extraída del vocabulario del espacio del problema o del espacio de la solución. Todo objeto tiene un nombre (se le puede identificar), un estado (generalmente hay algunos datos asociados a él) y un comportamiento (se le pueden hacer cosas a objeto y él puede hacer cosas a otros objetos).
- **Atributo:** Es una característica concreta de una clase (se refiere a las cualidades de la clase convertidas en variables)
- **Método:** Es una operación concreta de una determinada clase.
- **Instancia:** Es una manifestación concreta de una clase (un objeto con valores concretos).
- **Herencia:** Es un mecanismo mediante el cual se puede crear una nueva clase partiendo de una existente, se dice entonces que la nueva clase hereda las características de la clase existente aunque se le puede añadir más capacidades o modificar las que tiene.
- **Polimorfismo:** Hace referencia a la posibilidad de que dos métodos implementen distintas acciones, aun teniendo el mismo nombre, dependiendo del objeto que lo ejecuta o de los parámetros que recibe.

2.4 Diseño de sistemas con UML

¿Qué es UML?

UML (Unified Modeling Language, en español, lenguaje unificado de modelado) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos, ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

El modelado que se puede realizar con UML es muy amplio, y va desde el modelado de sistemas de información hasta aplicaciones distribuidas basadas en web.

UML es un lenguaje por que proporciona un vocabulario y las reglas para utilizarlo, además es un lenguaje de modelado lo que significa que el vocabulario y las reglas se utilizan para la representación conceptual y física del sistema.

UML es un lenguaje que nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño sin ninguna ambigüedad. En este contexto, UML sirve para especificar, modelos concretos, no ambiguos y completos.

En el siguiente diagrama podemos observar los elementos que conforman al lenguaje de modelado UML.

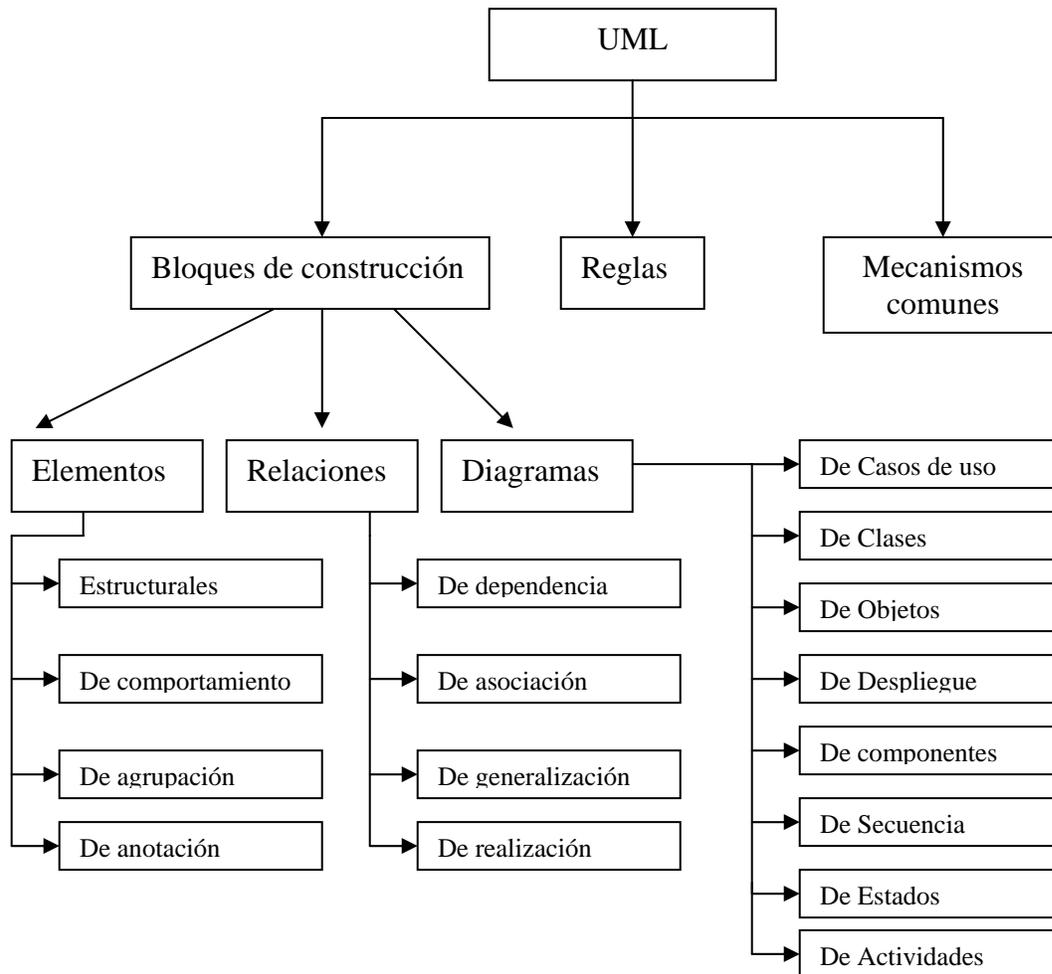


Figura 2.1: Elementos que conforman el UML.

El número de diagramas de que conste un diseño, depende de lo que queramos modelar, UML tiene una amplia gama de diagramas para definir un sistema.

UML tiene mecanismos los cuales ayudan a que alguna persona o entidad adapte el estándar a sus necesidades, las cuales son:

Especificaciones, que proporcionan una explicación textual de la sintaxis y la semántica de los bloques de construcción.

Las divisiones comunes permiten que los modelos se dividan en al menos un par de formas diferentes para facilitar la comprensión desde distintos puntos de vista.

Por último, los mecanismos de extensibilidad que UML proporciona sirven para evitar posibles problemas que puedan surgir debido a la necesidad de poder representar ciertos matices, por esta razón UML incluye los estereotipos, para poder extender el vocabulario con nuevos bloques de construcción, los valores etiquetados, para extender las propiedades un bloque, y las restricciones, para extender la semántica.

Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica.

Los diagramas estáticos son:

- Diagrama de clases: muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- Diagrama de objetos: Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- Diagrama de componentes: Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- Diagrama de despliegue.: Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.
- Diagrama de casos de uso: Muestran los casos de uso, actores y sus relaciones. Muestra quien puede hacer que y relaciones existen entre acciones(casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Los diagramas dinámicos son:

- Diagrama de secuencia, Diagrama de colaboración: Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.
- Diagrama de estados: muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.
- Diagrama de actividades: Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

Como podemos ver, el número de diagramas es muy alto, pero UML permite definir solo los necesarios, ya que no todos son necesarios en todos los proyectos.

2.5 Lenguaje Java.

2.5.1 Descripción del lenguaje de programación orientada a objetos: Java.

Java fue diseñado en 1990 por James Gosling, de Sun Microsystems, como software para dispositivos electrónicos de consumo. Curiosamente, todo este lenguaje fue diseñado antes de que diese comienzo la era World Wide Web, puesto que fue diseñado para dispositivos electrónicos como calculadoras, microondas y la televisión interactiva.

En los primeros años de la década de los noventa, Sun Microsystems decidió intentar introducirse en el mercado de la electrónica de consumo y desarrollar programas para pequeños dispositivos electrónicos. Tras unos comienzos dudosos, Sun decidió crear una filial, denominada FirstPerson Inc., para dar margen de maniobra al equipo responsable del proyecto.

Inicialmente Java se llamó Oak (roble en inglés), aunque tuvo que cambiar de denominación, debido a que dicho nombre ya estaba registrado por otra empresa. Se dice que este nombre se le puso debido a la existencia de tal árbol en los alrededores del lugar de trabajo de los promotores del lenguaje.

Tres de las principales razones que llevaron a crear Java son:

1. Creciente necesidad de interfaces mucho más cómodas e intuitivas que los sistemas de ventanas que proliferaban hasta el momento.
2. Fiabilidad del código y facilidad de desarrollo. Gosling observó que muchas de las características que ofrecían C o C++ aumentaban de forma alarmante el gran costo de pruebas y depuración
3. Enorme diversidad de controladores electrónicos. Java permite escribir un código común para todos los dispositivos.

Por todo ello, en lugar de tratar únicamente de optimizar las técnicas de desarrollo y dar por sentada la utilización de C o C++, el equipo de Gosling se planteó que tal vez los lenguajes existentes eran demasiado complicados como para conseguir reducir de forma apreciable la complejidad de desarrollo asociada a ese campo. Por este motivo, su primera propuesta fue idear un nuevo lenguaje de programación lo más sencillo posible, con el objeto de que se pudiese adaptar con facilidad a cualquier entorno de ejecución.

Basándose en el conocimiento y estudio de gran cantidad de lenguajes, este grupo decidió recoger las características esenciales que debía tener un lenguaje de programación moderno y potente, pero eliminando todas aquellas funciones que no eran absolutamente imprescindibles.

Bill Joy (cofundador de Sun y uno de los desarrolladores principales del sistema operativo Unix de Berkeley) juzgó que Internet podría llegar a ser el campo adecuado para disputar a Microsoft su primacía en el terreno del software, y vio en Oak el instrumento idóneo para llevar a cabo estos planes.

Para poder presentarlo en sociedad se tuvo que modificar el nombre de este lenguaje.

Java fue presentado en sociedad en agosto de 1995.

Algunas de las razones que llevaron a Bill Joy a pensar que Java podría llegar a ser rentable son:

- Java es un lenguaje orientado a objetos: Esto es lo que facilita abordar la resolución de cualquier tipo de problema.
- Es un lenguaje sencillo, aunque sin duda potente.
- La ejecución del código Java es segura y fiable: Los programas no acceden directamente a la memoria del ordenador, siendo imposible que un programa escrito en Java pueda acceder a los recursos del ordenador sin

que esta operación le sea permitida de forma explícita. De este modo, los datos del usuario quedan a salvo de la existencia de virus escritos en Java. La ejecución segura y controlada del código Java es una característica única, que no puede encontrarse en ninguna otra tecnología.

- Es totalmente multiplataforma: Es un lenguaje sencillo, por lo que el entorno necesario para su ejecución es de pequeño tamaño y puede adaptarse incluso al interior de un navegador.

2.5.2 Estructura de Java.

Java se compone de un compilador (javac), un intérprete (java) y una máquina virtual (JVM), el primer elemento se encarga de procesar el código java, es decir el lenguaje de alto nivel y cuyos archivos tienen la extensión .java, el compilador genera archivos tipo *bytes code*, .class, los cuales son procesados por el intérprete (java), posteriormente pasará nuestro programa al cargador de clases y por último es entregado a la máquina virtual, la cual ejecutará nuestro programa.

La máquina de bajo nivel o máquina virtual, es una máquina (stack, pila) abstracta concurrente, con una arquitectura y un reducido conjunto de instrucciones ligadas estrechamente al lenguaje, al grado de casi indicar el procesador en el cual se utiliza el lenguaje. Además describe un motor de ejecución con ciertos comportamientos como carga, arranque y apagado.

La carga de los módulos en Java, o clases, ocurre durante la ejecución del programa. Esto representa un cambio radical para los programadores acostumbrados al ciclo compilar-enlazar-cargar-ejecutar.

La carga de Java consta de varios pasos:

1. Ejecución de los bloques.
2. Verificación de que los trozos describen una clase bien estructurada que contiene código Java bien estructurado.
3. Construcción de una estructura de clase global.
4. Resolución de las referencias.
5. Control del acceso, permitiendo que una aplicación o entorno decida las reglas de acceso para la carga de la clase.

La siguiente figura muestra la forma en que trabaja el lenguaje java desde un archivo en código fuente (.java) hasta que es pasado a la máquina virtual.

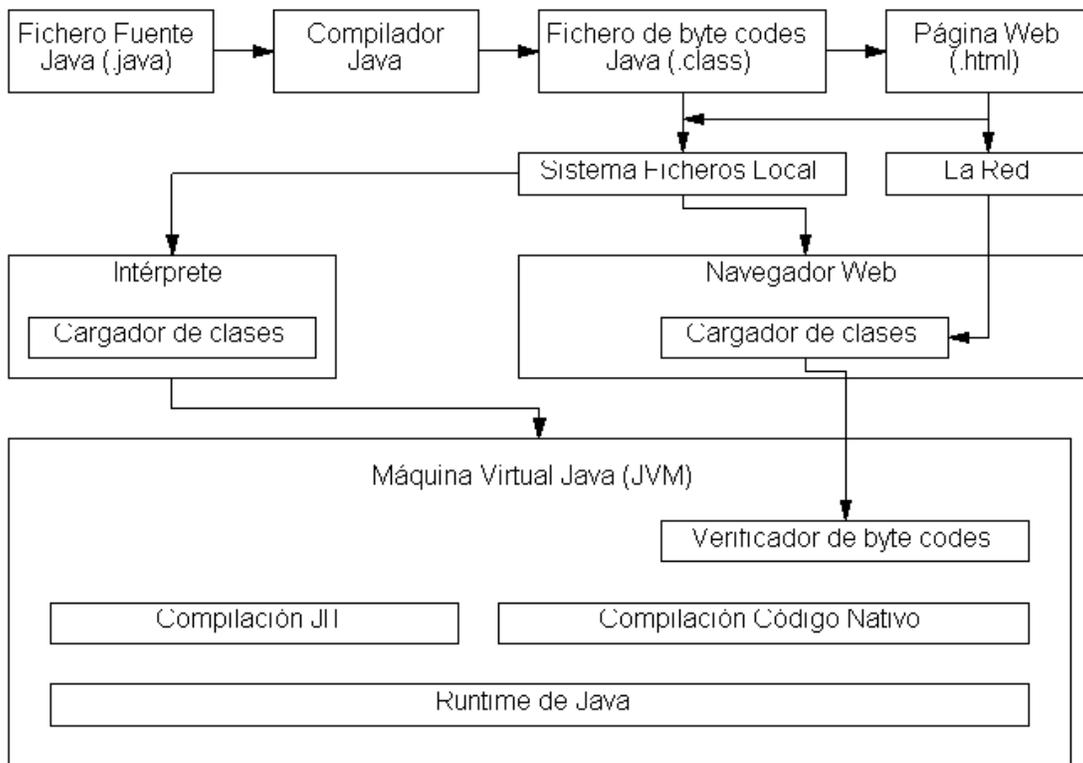


Figura 2.2 Entorno de ejecución de java.

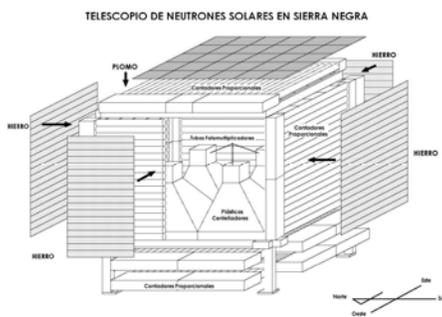
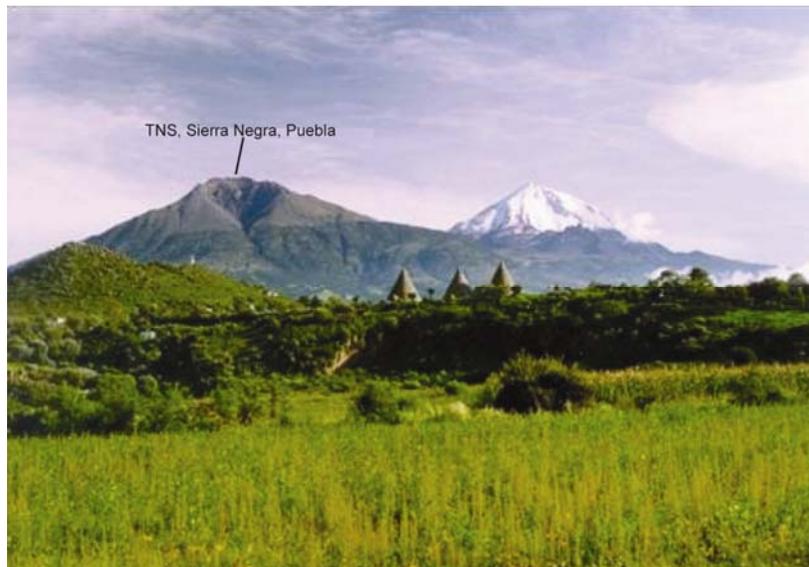
2.5.3 El entorno de aplicaciones java.

Aunque java no es un sistema operativo, debe proporcionar a las aplicaciones portables todo el entorno, desde servicios básicos de E/S para manipulación de cadenas, hasta interfaces de usuarios para trabajar en red. Java ha experimentado 3 actualizaciones principales, el kit de desarrollo inicio con el JDK1.0, siguió con el JDK1.1 hasta llegar al JDK1.5. en cada etapa Java fue incorporando nuevas herramientas hasta llegar, por ejemplo, a JFC Swing, que es un a aplicación GUI nativa de Java, modelos de imágenes 2D y 3D, etc.

Cuando se habla del JDK, se refiere al entorno de aplicación completo, es decir, a la máquina virtual de Java más las librerías de clases, cabe mencionar aquí que la máquina virtual de Java es la encargada de traducir las clases a lenguaje de máquina para poder ser ejecutadas, esto es lo que da la característica de multiplataforma a Java, ya que existe una distinta para cada sistema operativo (Windows, Linux, OS, etc).

Capítulo 3:

El Telescopio de Neutrones Solares de Sierra Negra, Puebla y Diseño del sistema.



Después de empaparnos de información acerca del Sol y de las herramientas de software, en esta sección se describe el punto de unión entre estos dos temas que a simple vista son totalmente diferentes y sin relación alguna, pero en esta sección comprenderemos la relación entre ambos temas.

3.1 El Telescopio de Neutrones Solares (TNS) de Sierra Negra, Puebla.

3.1.1 Antecedentes.

La detección de neutrones solares en la superficie de la Tierra nos provee de información importante acerca del mecanismo de aceleración de partículas de alta energía en la superficie del Sol. Un nuevo detector puede medir la energía y la dirección de llegada de los neutrones.

Los neutrones son producidos por fulguraciones solares, principalmente en las más fuertes. Estos eventos, que se describieron en el capítulo 1, son particularmente interesantes ya que conllevan información sobre el mecanismo de detección de iones dentro del ambiente de gas de la atmósfera solar. La detección de neutrones solares ha sido muy difícil, ya que casi el 50% de ellos se desintegran en el camino del Sol a la Tierra, además los neutrones son fuertemente atenuados en la atmósfera terrestre. La primera detección de neutrones solares fue realizada en 1980 por la misión SMM. La primera detección realizada en la Superficie terrestre fue hecha por los monitores de Jungfraujoch y Lomnický Stit el 3 de Junio de 1982, junto con observaciones satelitales. Ya se mencionaron algunos inconvenientes sobre la detección de neutrones en la Tierra, contrario a ello, se tiene la ventaja de que durante su viaje los neutrones no son afectados por campos magnéticos interplanetarios o turbulencia en el lugar de la fulguración. Por lo tanto los neutrones solares llevan información directa del sitio donde tuvo su origen la aceleración de partículas y ayudan a determinar si el proceso fue instantáneo o gradual. Por otra parte y hablando un poco más de las desventajas, los neutrones solares tienen masa, por lo cual, como ejemplo, los neutrones de 100 MeV de energía arriban 10 minutos más tarde que los de 1 GeV. Por lo tanto el saber el tiempo de aceleración de los iones es necesario para medir la energía de los neutrones, lo cual es una razón fundamental del uso de plásticos centelladores en el detector de Sierra Negra. Los neutrones son convertidos en protones dentro de los plásticos centelladores y la energía de los neutrones es medida por los protones producidos y que son recolectados, la energía de los protones es proporcional a la luz depositada en el centellador.

Ya se mencionó la observación del evento solar (fulguración) del 3 de Junio de 1982, para ella se tuvieron 2 interpretaciones, las cuales en los resultados finales no coinciden, las diferencias se deben a los modelos de interacción nuclear que se emplearon para explicar el fenómeno.

Los neutrones solares producen interacciones nucleares con la atmósfera terrestre, estas reacciones conducen a la atenuación de los neutrones, la cual debe ser tomada en cuenta en los equipos de detección. El 24 de mayo de 1990 se registró el evento más grande de neutrones solares reportado hasta el 2004, fue observado por varias estaciones del sector americano, incluyendo a la

estación de rayos cósmicos de la ciudad de México, el cual representó el segundo evento de gran flujo de neutrones detectado en esta estación, desafortunadamente este instrumento fue diseñado para obtener una información continua de rayos cósmicos galácticos, en donde el tiempo y resolución de energía y la dirección de arribo no sean críticos, por lo tanto no es el mejor detector para neutrones solares. Con los datos obtenidos de las diferentes del sector americano se realizaron los cálculos pertinentes arrojando resultados diferentes, por una parte los científicos que aplicaron el modelo de transporte de neutrones, a través de la atmósfera terrestre, “straight-ahead” obtuvieron una dispersión de neutrones de $\lambda = 208\text{g/cm}^2$ mientras que los que realizaron mediciones basándose en “cross-section” de dispersión de neutrones en carbono fue de $\lambda = 110\text{ g/cm}^2$. En un esfuerzo por aclarar las diferencias en los resultados se realizaron simulaciones y otro tipo de cálculos, la conclusión fue que se deben tomar en cuenta los efectos de refracción en la atmósfera terrestre.

Cabe mencionar que el TNS de Sierra Negra, pertenece a una red mundial de Telescopios de este mismo tipo (Telescopios de Neutrones Solares de Gran Altura), la figura 3.1 es muy explícita en cuanto a los miembros de la red mundial.

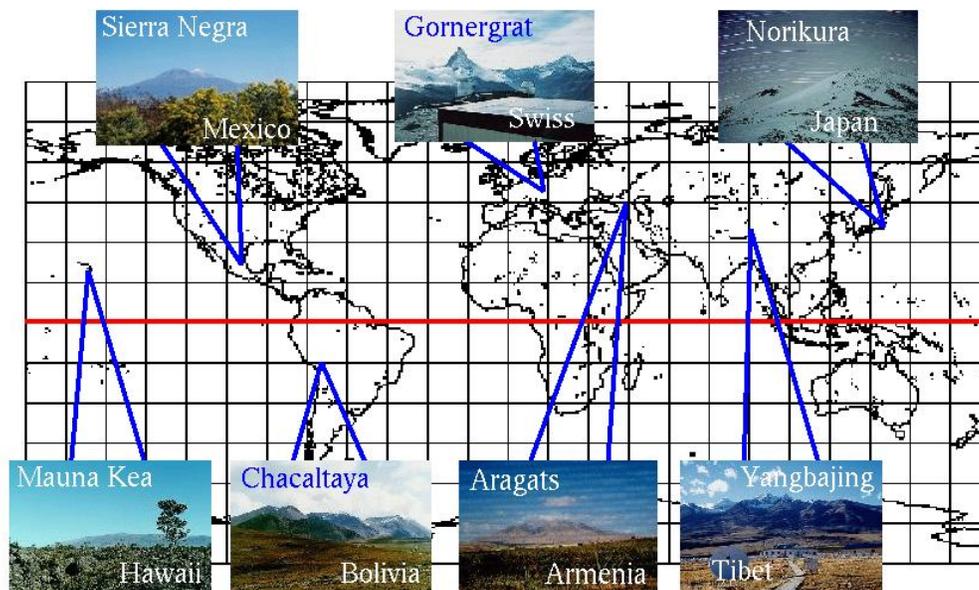


Figura 3.1: Red mundial de Telescopios de Neutrones Solares de gran altura.

3.1.2 El Telescopio de Neutrones Solares.

El TNS fue instalado y opera por la unión de esfuerzos del STELAB, de la Universidad de Nagoya en Japón y el Instituto de Geofísica, de la Universidad Nacional Autónoma de México.

El nuevo telescopio de neutrones solares instalado en Sierra Negra, a una altitud de 4580 m. sobre el nivel del mar y a una latitud de 19° N y 97.3°, es capaz de

registrar 4 umbrales de energía, $E > 30, 60, 90$ y 120 MeV, se conservó la idea del uso plásticos centelladores para la construcción de este nuevo telescopio. La energía de los neutrones entrantes es estimada por medidas de la luz emitida por los protones recolectados, producidos en las interacciones con el carbono y el hidrógeno dentro del centellador.

El telescopio de neutrones solares (TNS) consiste en un arreglo de 4 m^2 con centelladores plásticos de 30 cm de espesor. El detector esta rodeado por arreglos de contadores proporcionales para las anti-coincidencias. Por debajo del arreglo se colocaron dos capas de contadores proporcionales para determinar la dirección de arribo de los neutrones E-W y dos más para la dirección N-S. Un diagrama esquemático del detector se muestra en la figura número 3.2, por encima de los arreglos para las anti-coincidencias se colocó una placa de 5 mm de plomo donde 67% de los fotones incidentes se convierten en pares electrón-positrón. Para reducir la radiación de fondo por los lados de los contadores proporcionales fueron protegidos de cualquier radiación fotónica por placas de hierro de 10 mm de grueso.

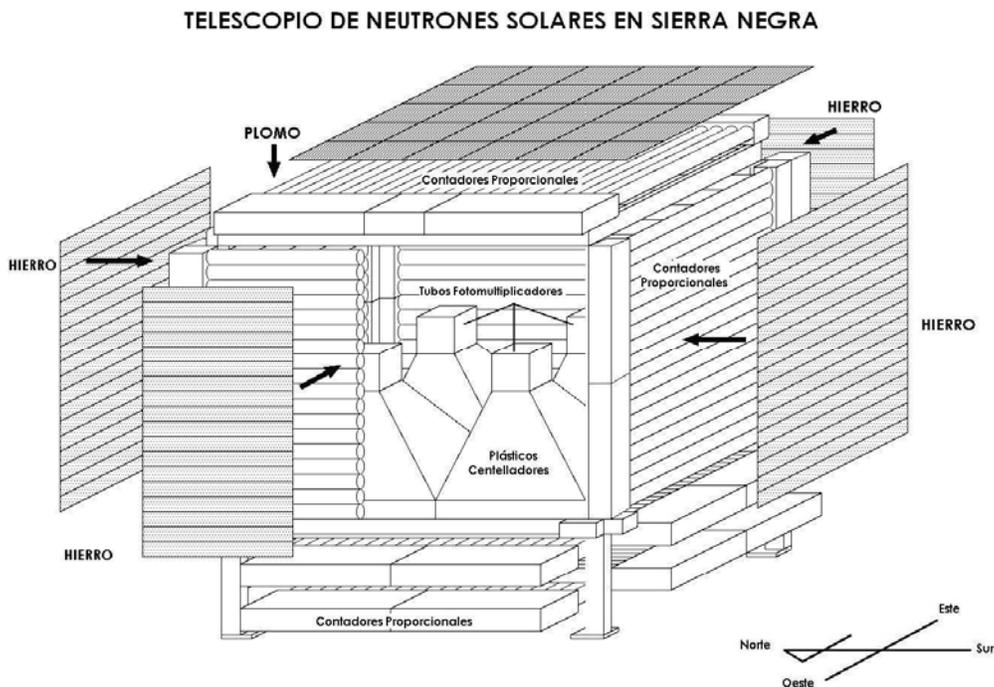


Figura.3.2. Vista esquemática del telescopio de neutrones solares de Sierra Negra.

Los plásticos centelladores se encuentran al interior de campanas pintadas de color blanco, arriba de cada campana hay un tubo fotomultiplicador, el principio básico de funcionamiento del detector nos dice que al chocar un neutrón con el plástico centellador se libera un protón en la misma dirección en que incidió el neutrón y a su vez se libera un fotón de tal forma que es detectado por el tubo fotomultiplicador, con ello se genera un pulso eléctrico, la intensidad de la luz, y por tanto la altura del pulso eléctrico producidos son proporcionales a la energía del neutrón incidente, lo cual nos permite conocer la energía del neutrón. A su vez

los contadores proporcionales cuentan partículas cargadas. El plomo en la parte superior y el hierro hacia los lados los detectores absorben los rayos γ , que pudieran producir señales similares a las de los neutrones.

En la parte de abajo existen contadores proporcionales que pueden detectar partículas cargadas, estas son las originadas en el plástico centellador a consecuencia de la interacción de los neutrones con el carbono. Los componentes que involucran a lo explicado en este último párrafo se pueden observar en la figura 3.3, así como los umbrales de energía que es capaz de detectar el telescopio.

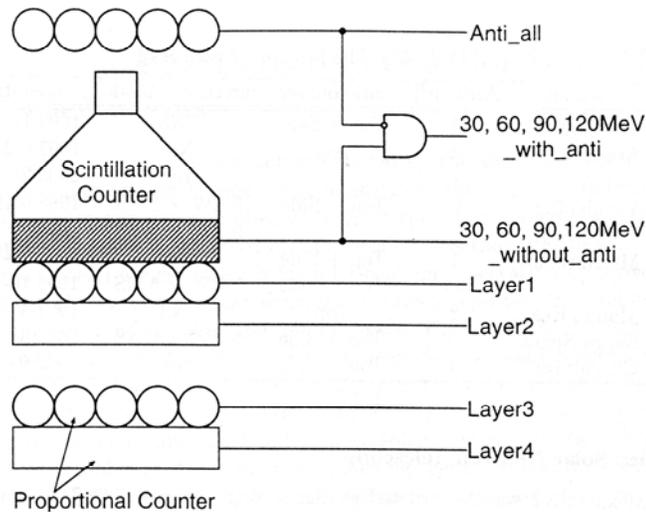


Figura.3.3: Sistema de detección de neutrones solares.

Se ha mencionado que se puede conocer la dirección de arribo de los neutrones, esto es gracias a los contadores proporcionales que se encuentran en la base del detector, ya se dijo que son 4, 2 para las direcciones E-W (Este-Oeste) y 2 para las direcciones N-S (Norte-Sur), la forma de determinar la dirección de los neutrones es de una forma indirecta, ya que la dirección es obtenida en base a los protones recolectados, los cuales son producidos por la interacción de los neutrones con el material centellador, en la figura 3.4, se aprecia como se da la detección de la dirección.

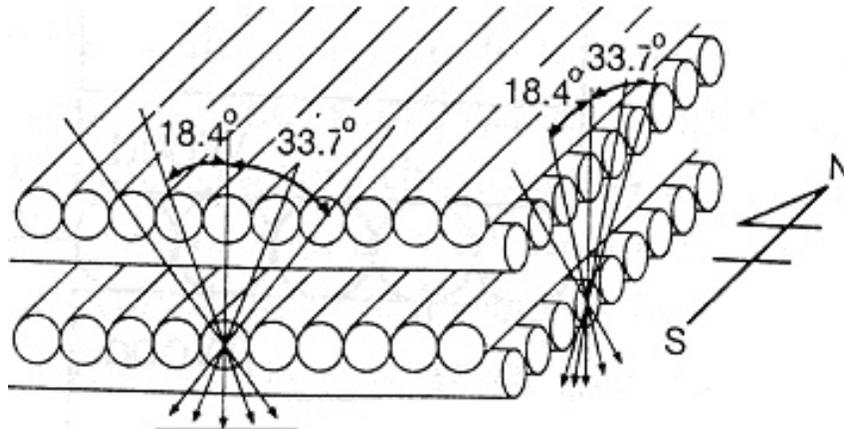


Figura 3.4: Dirección de arribo de los neutrones.

Este diseño provee una exactitud suficiente para estimar la dirección original de entrada de los neutrones con una exactitud de 15° , así se tiene la posibilidad de distinguir entre 25 posibles direcciones de arribo, por lo cual se ha realizado un mapa de direcciones, el cual es mostrado a continuación (figura 3.5):

EW4 NS3	EW2 NS3	EW0 NS3	EW1 NS3	EW3 NS3
EW4 NS1	EW2 NS1	EW0 NS1	EW1 NS1	EW3 NS1
EW4 NS0	EW2 NS0	EW0 NS0	EW1 NS0	EW3 NS0
EW4 NS2	EW2 NS2	EW0 NS2	EW1 NS2	EW3 NS2
EW4 NS4	EW2 NS4	EW0 NS4	EW1 NS4	EW3 NS4

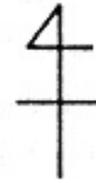


Figura.3.5: Mapa de dirección de arribo de los neutrones.

En este momento es pertinente numerar los canales detección del telescopio.

En general los canales detección pueden dividirse de la siguiente forma:
4 canales que detectan neutrones y 1 protones:

- Anti_all: detecta protones en todos los rangos de energía.

- S1 con anti: detecta neutrones con energía superior a 30 MeV.
- S2 con anti: detecta neutrones con energía superior a 60 MeV.
- S3 con anti: detecta neutrones con energía superior a 90 MeV.
- S4 con anti: detecta neutrones con energía superior a 120 MeV.

Los demás canales miden protones y direcciones en diferentes rangos de energía, un canal mide temperatura del medio y otro la presión, por facilidad se incluyen también en la lista los parámetros de año, mes-día, hora-minuto y segundo-milisegundo.

A continuación se listan los canales de detección:

Los canales son:

0) year	13) N-E	26) NS-ZW	39) layer2
1) month_day	14) N-ZE	27) NS-W	40) layer3
2) hour_min	15) N-EW	28) ZS-E	41) layer4
3) sec_msec	16) N-ZW	29) ZS-ZE	42) S1_and_anti
4) S1_with_anti	17) N-W	30) ZS-EW	43) muon1
5) S2_with_anti	18) ZN-E	31) ZS-ZW	44) muon2
6) S3_with_anti	19) ZN-ZE	32) ZS-W	45) muon3
7) S4_with_anti	20) ZN-EW	33) S-E	46) muon4
8) S1	21) ZN-ZW	34) S-ZE	47) press
9) S2	22) ZN-W	35) S-EW	48) temp
10) S3	23) NS-E	36) S-ZW	49) – 64) Otros
11) S4	24) NS-ZE	37) S-W	
12) Anti_all	25) NS-EW	38) layer1	

3.1.3 El sistema de cómputo.

El sistema adquirente de datos tiene dos componentes electrónicos esenciales, uno denominado CAMAC (contiene al cerebro de la detección de los datos el create controller), este es un módulo que sirve de interfaz entre el detector y la computadora en la que se almacenan los datos en formato ASCII. Un CPLD (Dispositivo Lógico Complejo Programable), se eligió este por cumplir con las expectativas de operación, tiene 2400 compuertas lógicas implementadas en su interior y 108 pines de entrada/salida, otra ventaja es que ocupa un espacio de 3 X 3 cm, este chip, grande en capacidad, se colocó en una tarjeta PCI, la cual va en el interior de la PC, a su vez esta tarjeta requiere de un adaptador y un cable plano que comunique a la tarjeta (por cierto el nombre comercial de la tarjeta es gpg-200 desarrollada por la compañía japonesa Interface) con el módulo CAMAC.

La plataforma de trabajo del sistema adquirente de datos es el sistema operativo con núcleo Linux, cuya distribución es Fedora, específicamente la versión 1.

A nivel de software, los elementos esenciales, son, obviamente los drivers de la tarjeta gpg-2000 y una serie de programas, en lenguaje C++, desarrollados por

científicos de Japón, dichos programas proveen de las facilidades de captura y almacenamiento de los datos que corresponden a las cuentas de neutrones detectados.

Los drivers de la tarjeta son llamados (ejecutados) cada vez que se inicia el sistema operativo, de igual forma es ejecutado el comando “daqstart” quien es el encargado de capturar los datos y almacenarlos en archivos de texto de codificación ASCII, el nombre archivo de texto tiene la siguiente estructura:

mxammdd.snx, en donde

aa: corresponde a 2 dígitos para el año.

mm: corresponde a 2 dígitos para el mes.

dd,: corresponde a 2 dígitos para el día

y en la extensión .snx,

x corresponde al número del archivo que se originó en el día, se pueden originar varios archivos si el sistema de adquisición de datos es detenido una o más veces en el transcurso del día.

Además, los investigadores de la Universidad de Nagoya, programaron algunas otras utilidades que facilitan el manejo, copiado y verificación de los datos que se generan en el sistema.

Una acotación importante es decir que se genera un archivo por día si no hay interrupciones en el sistema de adquisición de datos, y si existe interrupción se generará una serie de archivos con el mismo nombre pero con una extensión .snx, en donde x es el numero de archivo que se genero durante el día.

A continuación se hace una descripción más detallada de las herramientas de software que componen el sistema de adquisición de datos.

El sistema de software se compone de 6 comandos básicos, lo cuales son:

1. daqstart: es el comando que inicia la adquisición de datos, es el programa que carga el mayor peso, ya que es el encargado de tomar los datos y almacenarlos en un archivo de texto.
2. daqstop: este comando se ejecutará en una consola diferente a la cual se ejecuta el daqstart, y al ejecutarlo detendrá la adquisición de datos.
3. data: este comando muestra en pantalla el contenido del directorio en donde se guardan los archivos que se van generando.
4. cdw aa/mm/dd aa1/mm1/dd1: creará un archivo .tar.gz con los datos comprimidos del intervalo que abarca desde el día “aa/mm/dd” al día “aa1/mm1/dd1”
5. cdcheck: este comando nos muestra el contenido de nuestro cd.
6. cdclear: este comando nos ayuda a borrar discos regrabables.

También existe también un archivo de nombre “param”, archivo de texto, el cual nos permite modificar algunos parámetros para la adquisición de datos, como el intervalo de muestreo, el número de canales, y otros más.

3.2 Diseño del sistema.

Este apartado corresponde al diseño del sistema con el UML. Se presentan a continuación un análisis de requerimientos del sistema y su modelado con el UML. A partir del análisis de requerimientos obtuve los puntos esenciales que me ayudaron a lograr la construcción de los distintos diagramas de diseño con el UML.

3.2.1 Análisis de requerimientos.

Se requiere diseñar e implementar un sistema que procese, manipule y grafique los datos que se han obtenido, y se siguen obteniendo del Telescopio de Neutrones Solares de Sierra Negra, Puebla.

Los datos que se tienen actualmente se encuentran en dos formatos diferentes debido a la actualización del sistema de cómputo, por lo cual, para datos anteriores al 2005 se trabajará con archivos de las siguientes características:

1. Fueron generados por un sistema UNIX Solaris.
2. Tienen un formato de texto basado en codificación ASCII.
3. Están comprimidos con el compresor gzip de UNIX Solaris.
4. Tienen extensión .sn*, donde * indica el número de archivo consecutivo generado en un día.
5. Presentan (ya descomprimidos), una distribución de 65 columnas y 8,640 filas (en total del muestreo para un día entero), en donde cada fila corresponde al muestreo de cada 10 segundos, que realiza el TNS.

El segundo formato de datos sobre el que se trabajará tiene las siguientes características:

1. Son generados por un sistema Linux Fedora Core 1.
2. Tienen un formato de texto basado en codificación ASCII.
3. Los archivos ya no se encuentran comprimidos
4. Tienen extensión .sn*, donde * indica el número de archivo consecutivo generado en un día.
5. Presentan una distribución de 54 columnas por 8645 filas, en donde las primeras 5 filas solo tiene una columna y son los datos de algunos parámetros de configuración, posteriormente viene los datos propiamente hablando.

Los datos se obtienen por diversas vías: medio de almacenamiento (CD, Disquete, Disco Duro, etc.). El usuario requiere gráficas de los datos con las siguientes resoluciones:

1. 1 minuto
2. 3 minutos
3. 5 minutos
4. 1 hora

El archivo original almacena datos con resolución de 10 segundos.

Las resoluciones requeridas resultan de la acumulación de los de menor resolución, por ejemplo para hacer el cálculo del dato correspondiente al primer minuto del día, se deberán sumar los 6 primeros datos de 10 segundos, para generar el primer dato del día de 3 minutos se deberán sumar los 3 primeros datos de 1 minuto, y así sucesivamente; este procedimiento aplica para todos los canales de datos, exceptuando los referentes a presión y temperatura, ya que para estos, los cálculos deberán hacerse obteniendo promedios.

Esto permitirá delimitar la información que al usuario le interese en un determinado momento, teniendo como punto central el poder visualizar una gráfica o tabla de los datos de interés. Es decir, el usuario introducirá vía una interfaz gráfica, los siguientes elementos:

1. Fecha
2. Resolución de los datos (10 seg., 1min. 3min., 5min., 1hora, ó 1día)
3. Hora de inicio, a partir de la cual quiero visualizar mi gráfica.
4. Hora final, hasta donde quiero que termine mi grafica.
5. Modo de adquisición de los datos (local o vía servidor).

Una vez que el usuario ingrese estos requerimientos, atendiendo una sintaxis específica para cada elemento, el usuario podrá pedir al sistema que le muestre una tabulación de los datos y/o una gráfica de los 4 canales principales (S1 with anti, S1, Anti_all y Muon1).

Además de esto el usuario podrá salvar los datos graficados, los datos de un día de alguna resolución en específico, los datos de todas las resoluciones para un día entero. De igual manera el usuario deberá poder salvar la última gráfica que generó el sistema en tres formatos diferentes .tex, .ps, y .png. Adicionalmente el usuario podrá solicitar al sistema gráficas de otros canales de datos y también una gráfica de los 4 canales principales que abarquen 2 días, cuatro graficas máximo a la vez.

Como consideración adicional el usuario debe tener la libertad de borrar mediante la interfaz gráfica el contenido del directorio temporal en donde se almacenaron los datos necesarios para satisfacer las peticiones del usuario.

Se plantea, por otra parte, la necesidad de implementar el sistema con software libre, y utilizar plataforma Linux (específicamente Fedora Core 1), además de que las gráficas sean generadas con GNUPlot, que es un software de graficación libre. Adicionalmente decidí desarrollar el procesamiento de los datos así como la interfaz gráfica en lenguaje Java, ya que provee de las herramientas necesarias para implementar un sistema orientado a objetos, además provee de facilidad para crear interfaces gráficas (con AWT o Swing) y por supuesto del manejo de archivos.

3.2.2 Diseño del sistema con UML.

En el capítulo anterior se describieron las características más importantes de este lenguaje de modelado, habiendo estudiado la simbología y la lógica del lenguaje se procede a mostrar y explicar los diagramas elaborados.

3.2.2.1 Diagrama de Casos de uso.

Este diagrama muestra interacciones entre el usuario y la interfaz gráfica (ver figura 3.6a), y además muestra interacciones entre la interfaz gráfica y algunos elementos del sistema (ver figura 3.6b), en este caso particular el sistema cuenta con un solo actor denominado usuario, que es quien utilizará el sistema para generar gráficas o tablas de su interés.

Actores:

- 1) **Usuario:** Es el actor cuyo papel es desarrollado por la persona que va a hacer uso del sistema, es decir, algún investigador o estudiante el cual tenga por objetivo realizar un análisis de los datos mediante gráficas y tablas.

- 2) **Interfaz Gráfica:** Se considera como actor, sobre todo en el procesamiento de los datos y la salida de los mismos.

Lista de Casos de Uso:

- 1) *Solicita gráfica de un día o fracción, de 4 canales:* Este caso de uso se aplica cuando el usuario solicita al sistema en la pantalla principal generar una gráfica de 4 canales de datos de un día o fracción.

- 2) *Solicita gráfica de 2 días de 4 canales:* Este caso de uso es aplicado cuando el usuario desea visualizar en pantalla una gráfica de dos días de datos contiguos de los 4 canales principales de datos.

- 3) *Solicita tabla de 4 canales:* Este caso de uso se aplica cuando el usuario pide al sistema la generación de una tabla de 4 canales de datos.

- 4) *Solicita tabla de todos los canales:* Este caso de uso se aplica cuando el usuario pide al sistema una tabla con los datos de todos los canales.

- 5) *Guardar datos de 4 canales:* Este caso de uso se aplica cuando el usuario desea guardar los datos de 4 canales principales de la última gráfica o tabla mostrada.
- 6) *Guardar datos de todos los canales:* Este caso de uso se aplica cuando el usuario desea guardar los datos de los 45 canales correspondientes a un día determinado y que van de acuerdo a la última gráfica o tabla generada por el sistema.
- 7) *Guardar datos de todas las resoluciones:* Este caso de uso se emplea cuando el usuario requiere salvar en alguna ubicación específica los datos de todas las resoluciones correspondientes a la fecha que corresponde a la última gráfica o tabla generada.
- 8) *Guardar la última gráfica generada:* Este caso de uso se emplea cuando el usuario pide al sistema salvar en alguna ubicación específica la última gráfica generada por el sistema.

Existen una serie de casos de uso, cuyo actor lo nombre como Interfaz Gráfica, lo que quiero expresar con este actor es la interacción que guarda el propio sistema (mediante la interfaz gráfica) con un conjunto de subsistemas o programas cuya relación es importante destacar.

El actor Interfaz Gráfica tiene los siguientes casos de uso:

- 1) *Procesar Datos:* El sistema pide que se procesen los datos que solicitó el usuario.
- 2) *Copia datos desde el origen:* Este caso de uso se emplea cuando el sistema requiere que los datos sean copiados a un directorio temporal para su manejo.
- 3) *Descomprimir datos:* Existen datos en formato comprimido, cuando se presenta este caso se lleva a cabo el caso de uso presente.
- 4) *Generar archivos de todas las resoluciones:* Es el caso de uso general para generar los archivos de uno o de 2 días de datos.
- 5) *Generar archivos de un día:* Una vez capturados los datos del “archivo crudo” se procede a generar archivos de un día de datos de distintas resoluciones, primero un archivo de 45 canales para luego generar uno de 4 canales (los 4 canales deseados).
- 6) *Generar archivo de 2 días:* Este caso de uso se emplea cuando se requiere una gráfica de 2 días, entonces el sistema hace una petición para generar un archivo de 2 días con los 4 canales principales.

- 7) *Mostrar salida de datos*: Es caso de uso que define la salida de datos del sistema ya sea una tabla o una gráfica.
- 8) *Mostrar gráfica*: El sistema solicita un subprocesamiento para que sea mostrada la gráfica que pidió el usuario.
- 9) *Mostrar tabla*: El sistema hace una petición a un subprocesamiento para generar una tabla de los datos que solicitó el usuario y mostrarla en pantalla.

3.2.2.2 Diagrama de paquetes.

Los paquetes (ver figura 3.7) que se emplearon fueron los siguientes:

IG: Este paquete contiene las clases en las cuales se contemplan elementos de interacción con el usuario, ventanas, paneles, campos de texto, etc.

Datos: Este paquete contiene todas las clases referentes al manejo de los datos para generar un archivo de una determinada resolución, de cierta fecha, a partir de los datos crudos.

Graficas: Este paquete con tiene la clase cuya función es generar el script de extensión .gpl (gnuplot) para poder generar la gráfica solicitada por el usuario.

Utilerias: Este paquete contiene algunas clases de apoyo para el adecuado procesamiento de los datos, por ejemplo, contiene un método de ordenación, contiene casi todas las cadenas de texto que nunca cambian en el programa, como son los rótulos de las etiquetas, así como la ruta del sistema, entre otras cosas.

3.2.2.3 Diagrama de clases.

Este diagrama (ver figura 3.8) muestra todas las clases empleadas en el sistema y las relaciones que hay entre ellas, además de la cardinalidad de la relación entre clase y clase. Es un diagrama de clases sencillo que ilustra únicamente a las clases por nombre sin listar métodos ni variables, la explicación del contenido de código de cada clase se hace en otro apartado.

Lista de clases:

- | | |
|-------------------------|----------------------------|
| 1) InterfazGrafica. | 8) ProcesaDatos |
| 2) GraficaOtrosCanales. | 9) Dato4canales |
| 3) Grafica2Dias | 10) Dato45canales |
| 4) ContadorGraficas | 11) Obtenerdatos |
| 5) EjecutaComando | 12) Grafica |
| 6) Parámetros | 13) GeneraArchivoNecesario |
| 7) ProcesamientoDeDatos | |

3.2.2.4 Diagrama de actividades.

Este Diagrama (ver figura 3.9a y 3.9b) muestra las diferentes acciones (más importantes) y decisiones que lleva a cabo el sistema para lograr una salida (gráfica o tabla).

El arranque del sistema lo marca la invocación de este mediante la llamada al sistema en una línea de comandos, posteriormente el usuario ingresa una serie de datos que requiere el sistema, el siguiente paso es que el usuario solicita una gráfica o tabla, en base a lo que el usuario ingreso en la ventana principal de la interfaz gráfica, se busca si existe el archivo de apoyo para la graficación o la generación de la tabla en el directorio temporal, si existe el archivo es por que en algún momento dado ya se generó tal archivo, después solamente se verifica el tipo de salida de datos que solicitó el usuario y se muestra en pantalla (Una tabla o una gráfica), posteriormente el usuario tiene la posibilidad de pedir una nueva salida de datos, con otros parámetros, cerrar la aplicación o elegir algún elemento del menú de opciones, todo esto si el archivo fue encontrado en el directorio temporal. Cuando el archivo no exista en el directorio temporal se desencadena el procesamiento de los datos propiamente, iniciando con su obtención desde una ubicación local o remota, posteriormente se verifica si los datos requieren descompresión posteriormente se procede a generar los archivos de las diferentes resoluciones, una vez teniendo estos archivos se genera otro de cuatro canales, el cual está listo para graficarse o tabularse, mostrada la salida de los datos el usuario tiene la opción de elegir de nueva cuenta entre solicitar otra gráfica o tabla de otra fecha u otra resolución o bien elegir una opción del menú, y si no puede escoger salir del sistema, es decir, cerrar la ventana principal.

El menú de opciones puede desencadenar varias actividades, para generalizar se describen de la siguiente forma:

- Actividades de para salvar archivos: El usuario puede salvar los archivos de 4 canales, de todos los canales, y de todos los canales y todas las resoluciones., además de poder guardar la última gráfica generada.
- Actividades para mostrar tablas: El usuario puede solicitar ver tablas de 4 canales o de 45 canales.
- Actividades adicionales de graficación: El usuario puede pedir mediante estas actividades la graficación de cualesquiera otros canales o bien una gráfica de 2 días de los 4 canales principales.

3.2.2.5 Diagrama de secuencia.

El diagrama de secuencia (ver figura 3.10a y 3.10b) corresponde a la descripción que se hace del sistema en la sección de Descripción del funcionamiento del sistema a nivel de subprogramas u objetos.

3.3 Conclusiones del diseño.

Los diagramas en conjunto muestran diferentes vistas del funcionamiento y la estructura de nuestro sistema, una vez que se inició el diseño con el análisis de requerimientos, después de un análisis detallado, se desprendió la realización de los distintos diagramas, en donde podemos distinguir las diferentes vistas.

En general en este apartado se aplicó la teoría referente al UML, se diseñó un sistema de procesamiento y graficación de datos de acuerdo a las necesidades planteadas por el Dr. José Francisco Valdés Galicia en torno al manejo de los datos del TNS (Telescopio de Neutrones Solares) de Sierra Negra Puebla. Una vez culminado el diseño, el siguiente paso es la programación del sistema basado en el diseño obtenido.

Los diagramas facilitarán la programación y las subsecuentes adecuaciones o modificaciones, lo cual permitirá en el futuro ampliar, modificar o reducir los componentes del sistema de una manera más sencilla, comparada a que si los diagramas no hubieran sido elaborados.

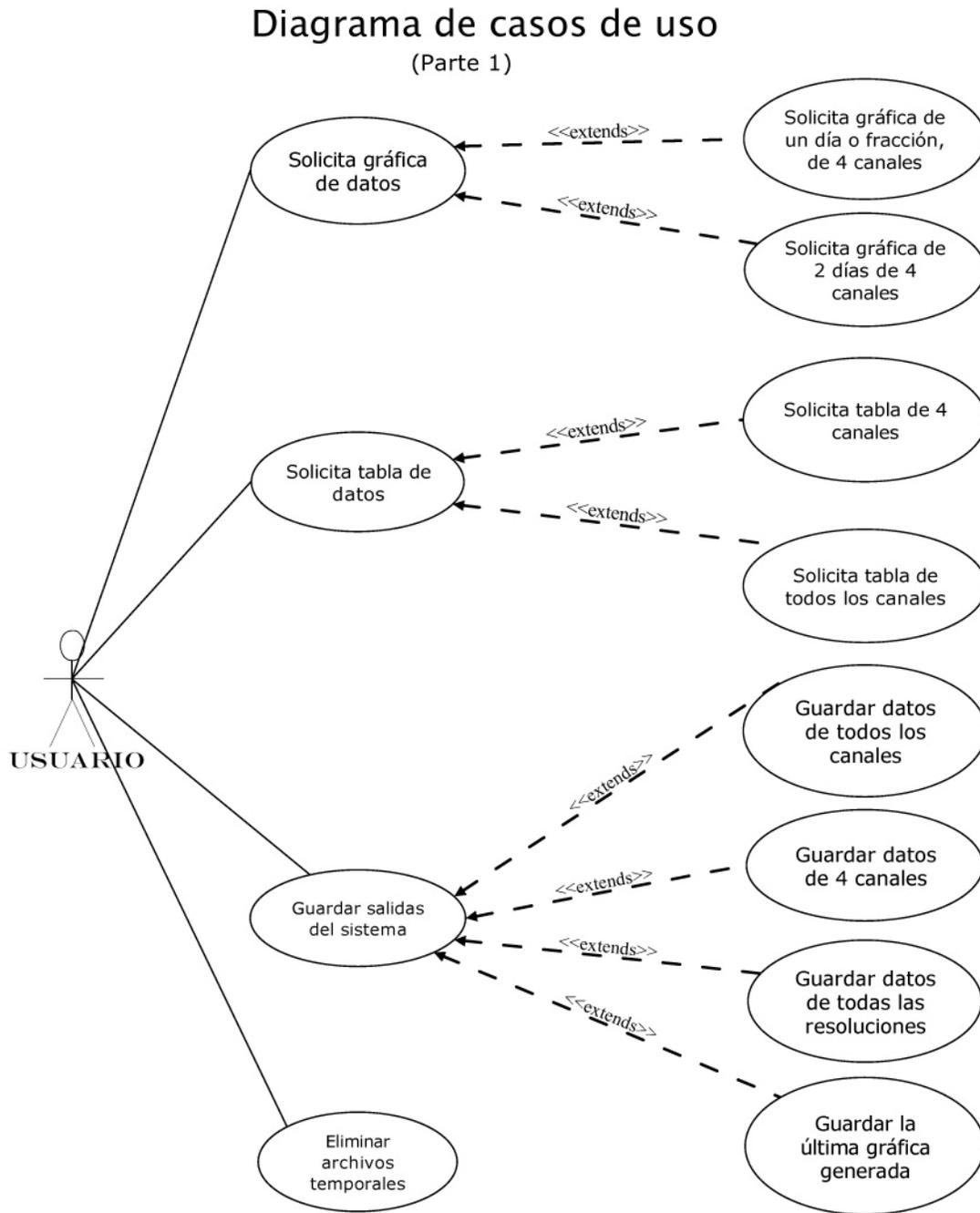


Figura 3.6a: Diagrama de casos de uso, parte 1.

Diagrama de casos de uso (Parte 2)

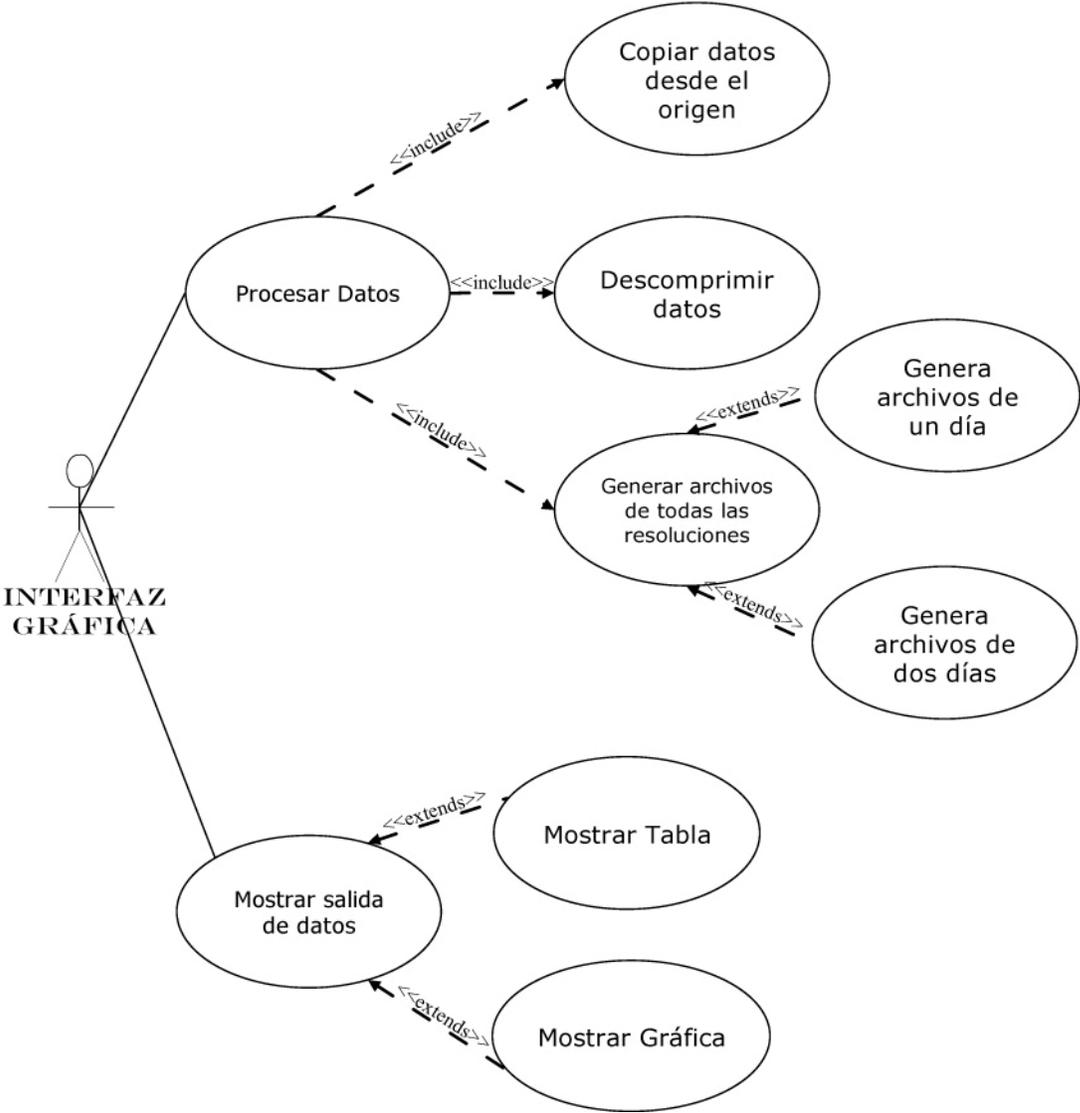


Figura 3.6b: Diagrama de casos de uso, parte 2.

Diagrama de paquetes

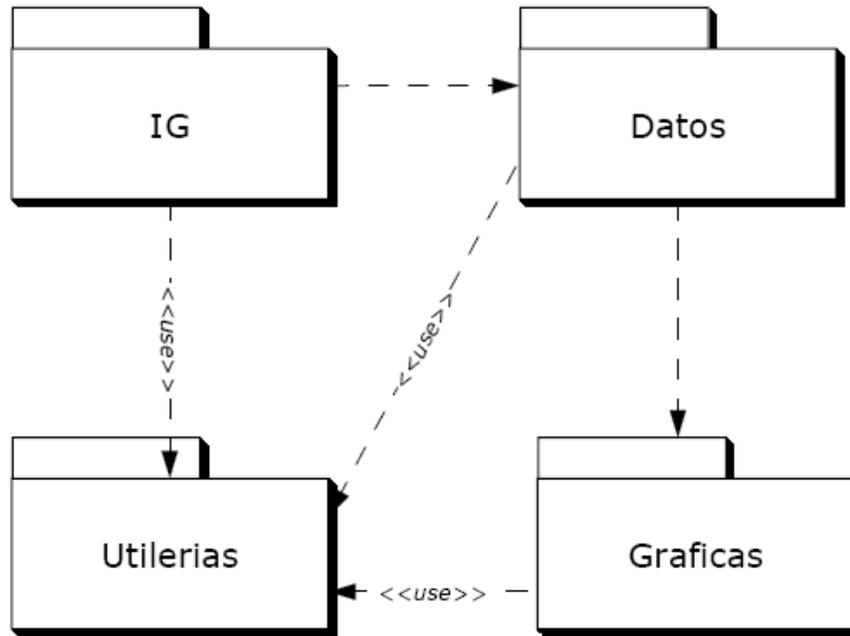


Figura 3.7: Diagrama de paquetes.

Diagrama de clases

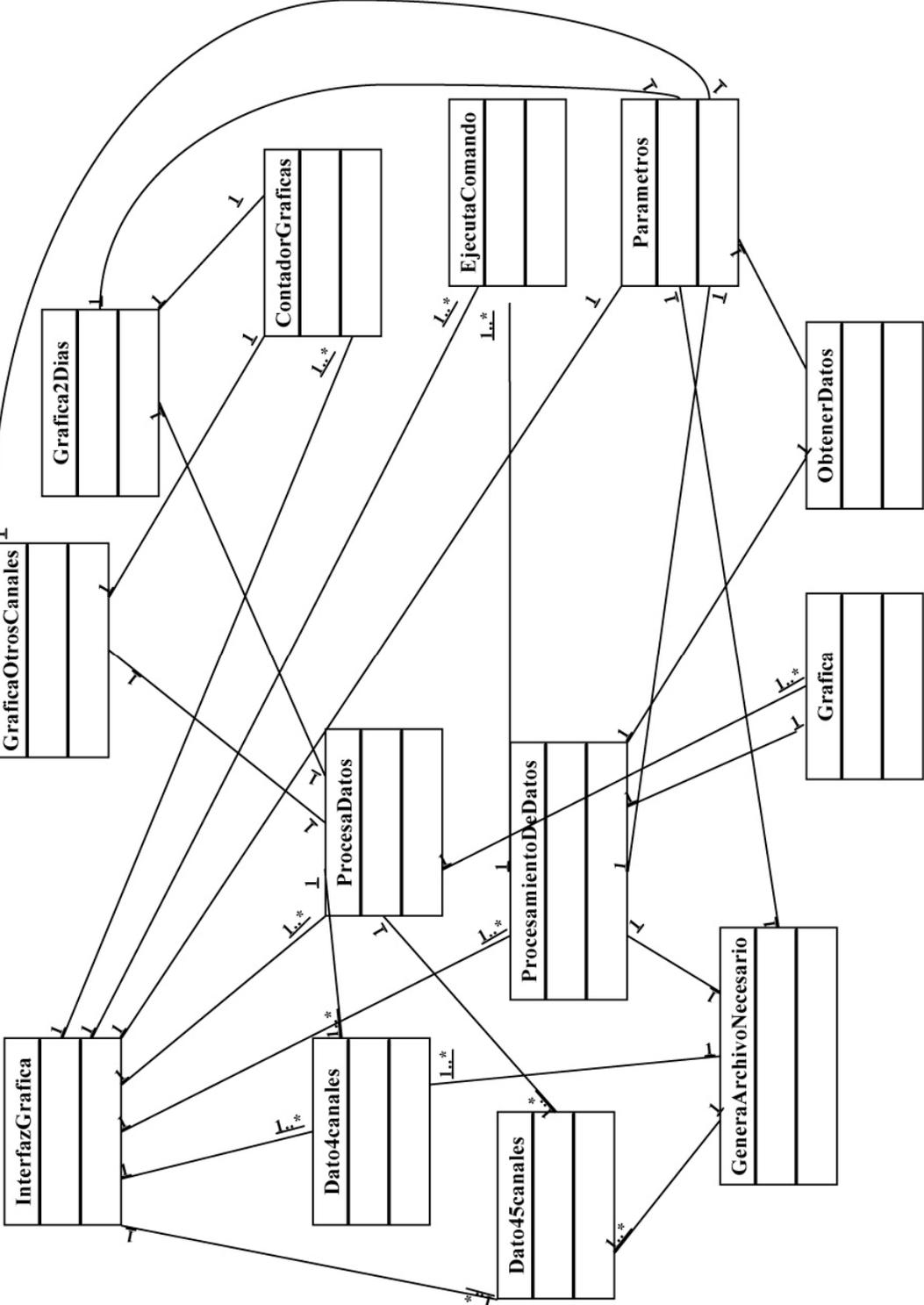


Figura 3.8: Diagrama de Clases.

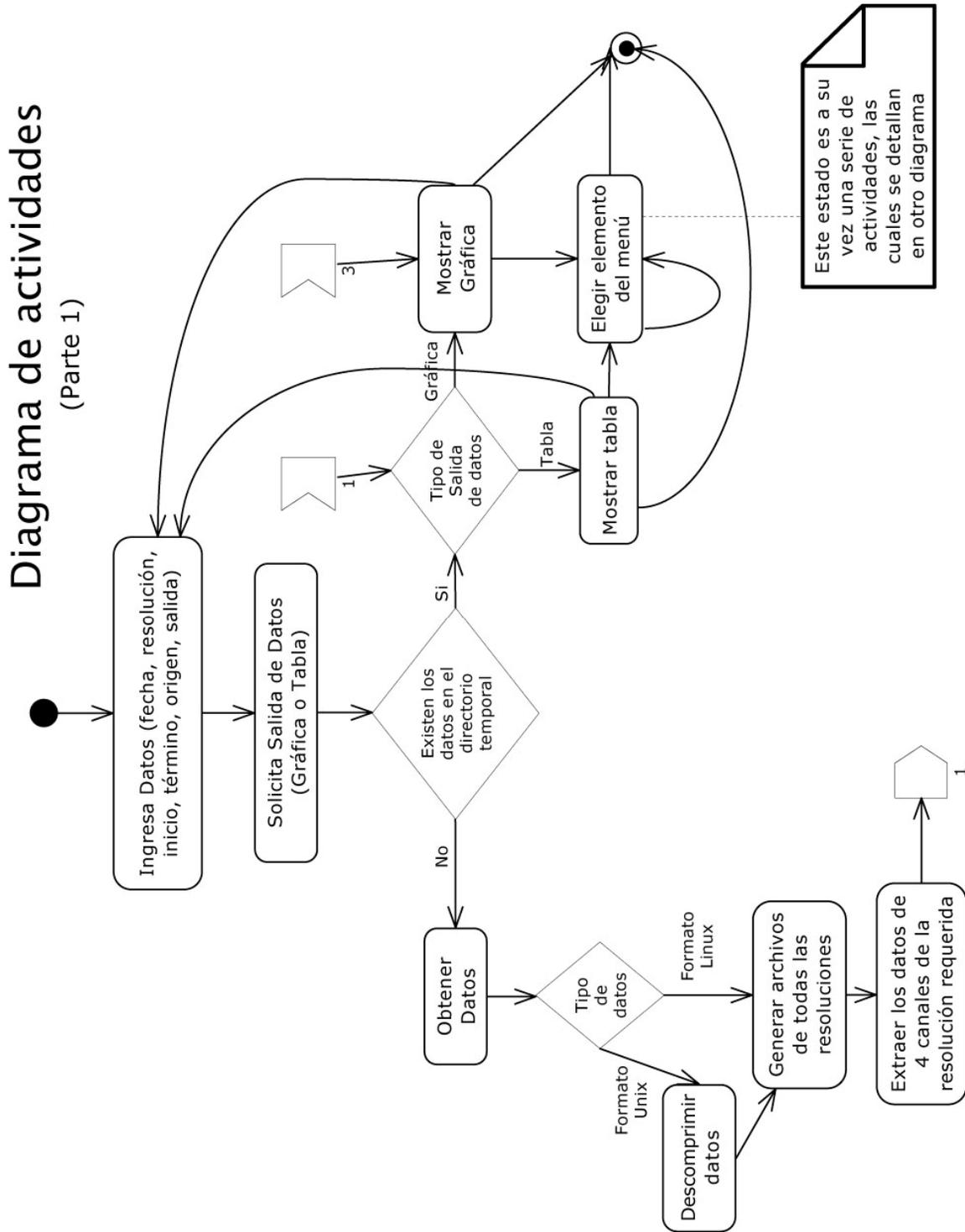


Figura 3.9a: Diagrama de actividades, parte 1.

Diagrama de actividades

(Parte 2)

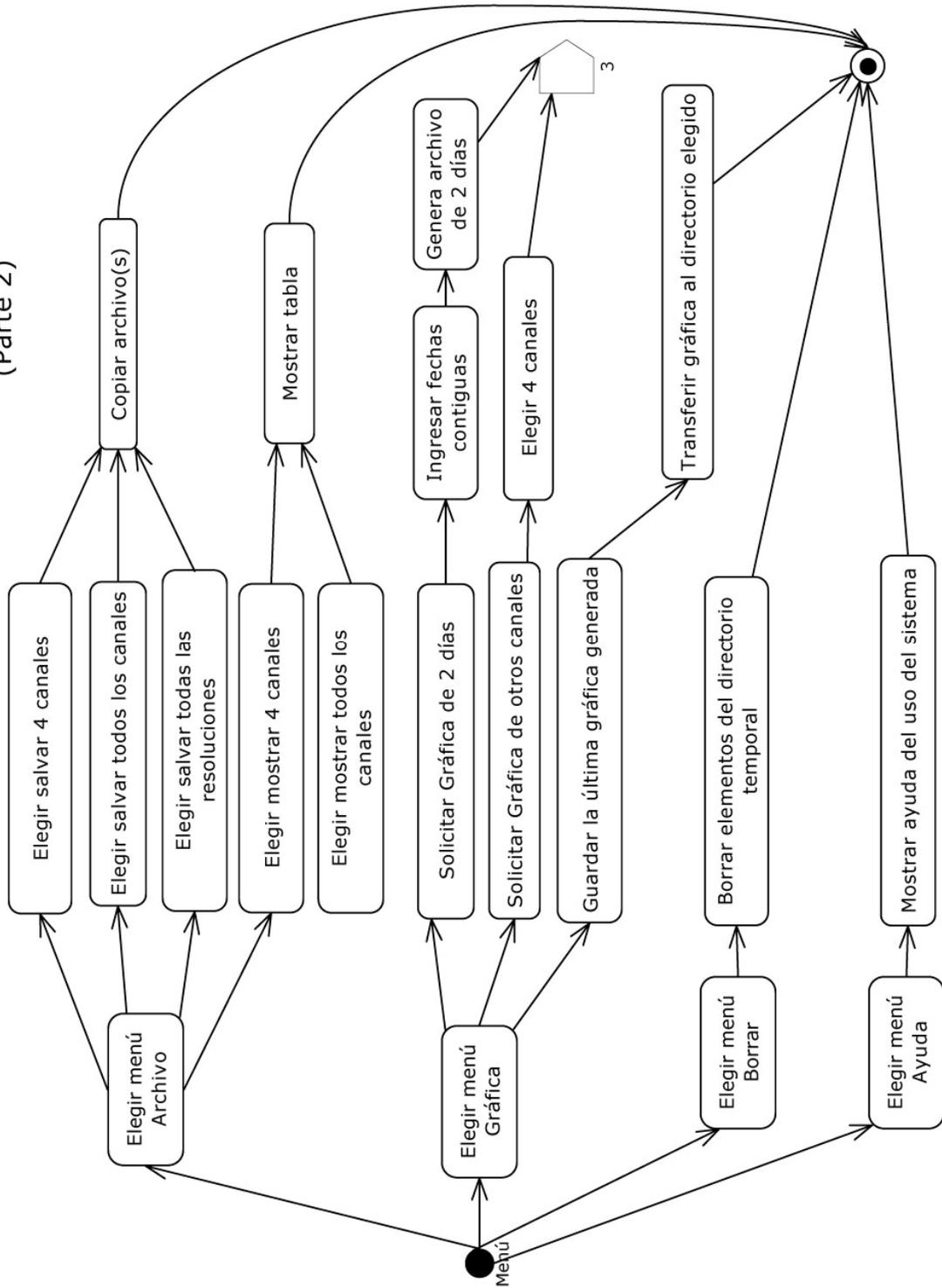


Figura 3.9b: Diagrama de actividades, parte 2.

Diagrama de secuencia

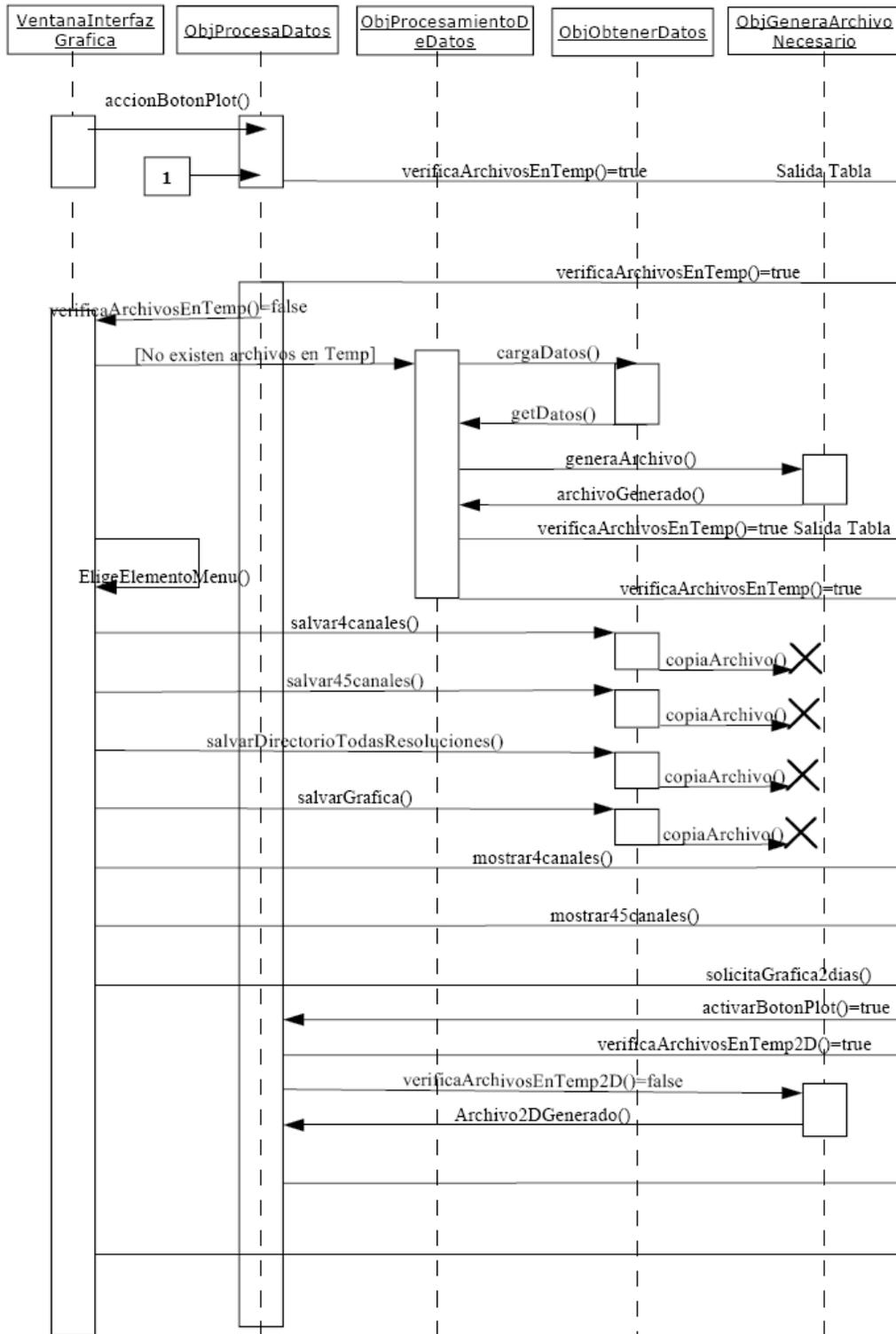


Figura 3.10a: Diagrama de Secuencia, parte 1.

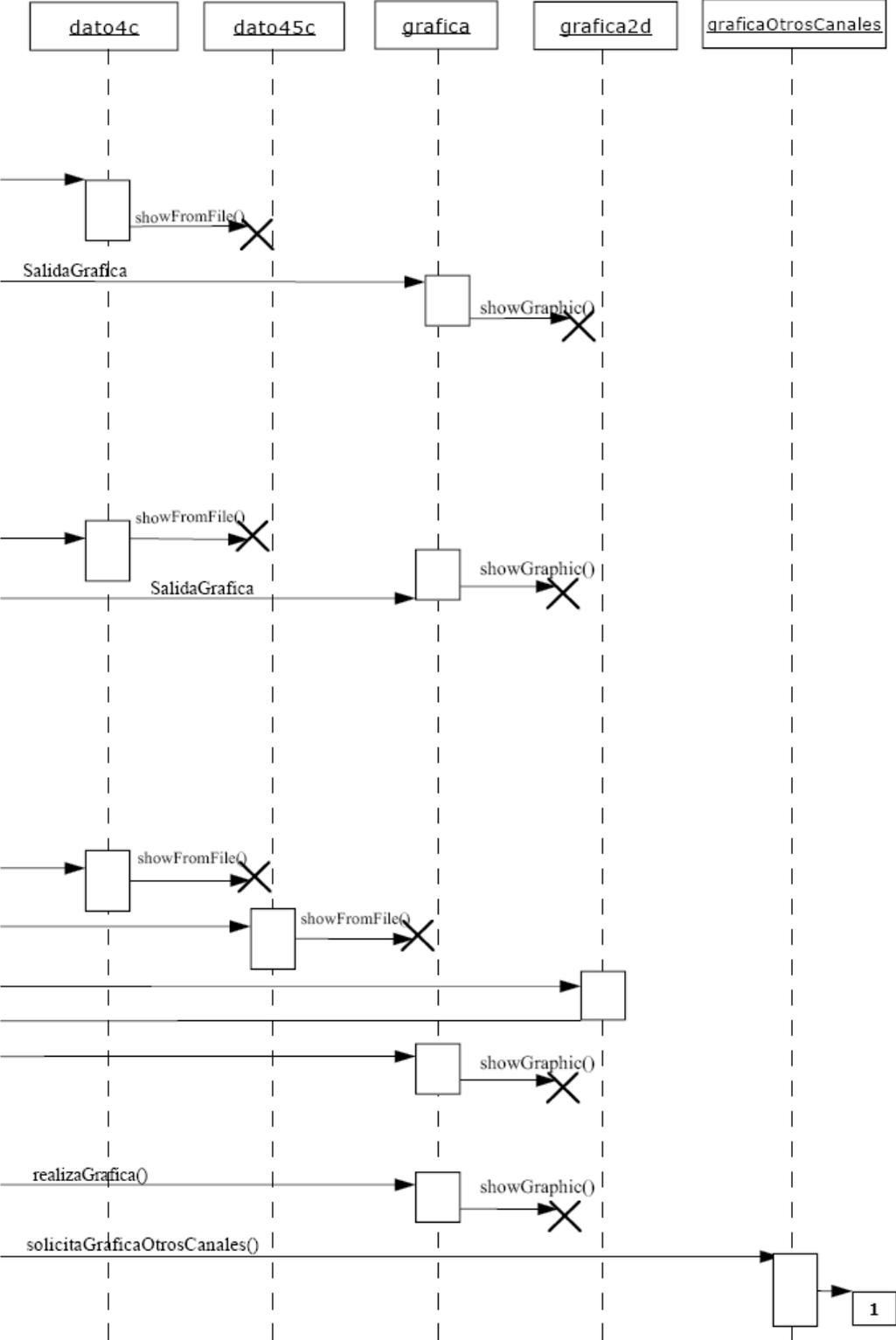


Figura 3.10b: Diagrama de Secuencia, parte 2.

Capítulo 4.

El Sistema: Programación, estructura y funcionamiento.



Una vez terminado el diseño proseguí con la programación del sistema, como ya se mencionó antes, la programación se llevó a cabo en el lenguaje de programación orientado a objetos java. En esta sección se describe la estructura y el funcionamiento del sistema.

4.1 Descripción general de la estructura del sistema.

Estructura del sistema a nivel directorios, clases y archivos temporales.

Se decidió nombrar al sistema como: **SNGraphics**, y está conformado de una serie de clases programadas en lenguaje java; el sistema está pensado para su funcionamiento en plataforma Linux, java es independiente de la plataforma, por eso, se considera fácil su futura adaptación a otros sistemas operativos.

El sistema se instala en el directorio /usr/local en una carpeta llamada sng, la cual contiene a su vez los siguientes directorios:

- 1) **System**: Contiene los 4 paquetes del sistema (ver figura 4.1):
 - a) **Datos**: Contiene todas las clases que se encargarán del procesamiento de los datos (Copia, obtención, cálculos, guardado y visualización de datos).
 - b) **Graficas**: Contiene la clase que se encargará de generar y ejecutar el script gnuplot (.gpl) de acuerdo a la solicitud hecha por el usuario.
 - c) **IG**: Contiene todas las clases relativas a la interfaz gráfica del sistema, es decir las clases que se encargaran de la interacción con el usuario.
 - d) **Utilerias**: Se compone de una serie de clases que sirven de apoyo para la correcta realización de algunas acciones en el sistema y su principal objetivo es evitar la redundancia en el código.

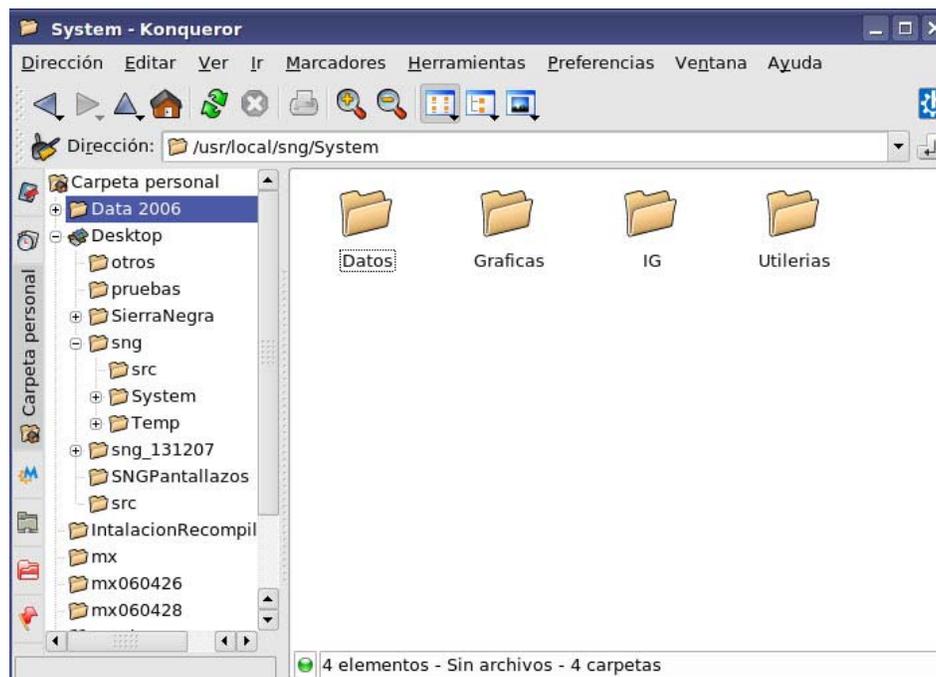


Figura 4.1: Paquetes del sistema.

- 2) **Temp**: Este directorio sirve de apoyo al sistema en la generación de archivos y gráficas temporales, los temporales sirven para mostrar al usuario las salidas del sistema.

4.2 Descripción general del funcionamiento del sistema.

4.2.1 Ejecución del sistema.

Una vez instalado el sistema **sngraphics** (véase Manual de instalación), se puede ejecutar abriendo una línea de comandos, en la cual hay que escribir el nombre del comando: `sngraphics`, y presionar la tecla enter (ver figura 4.2).

4.2.2 Ventana principal.

Posteriormente, se muestra en nuestro monitor la ventana principal del sistema, que sirve para la interacción con el usuario, se muestra la ventana principal en la figura 4.3.

En esta ventana se pueden apreciar todos los elementos que el usuario debe ingresar para generar una gráfica o una tabla de datos.

Como se muestra en la imagen el usuario debe ingresar la fecha en formato `aa/mm/dd`, la resolución con la que desea visualizar la salida de datos, esta puede ser de 10 segundos, 1 minuto, 3 minutos, 5 minutos y 1 hora, por defecto, si no se ha elegido alguna resolución, se toma la resolución de 1 minuto.

El usuario también debe elegir el intervalo de tiempo, en el cual desea observar la salida de datos (gráfica o tabla); así mismo el usuario debe elegir entre dos opciones para que el sistema realice una copia de los datos para poder trabajar con ellos, la forma local adicionalmente requiere que se le indique al sistema la ubicación absoluta (ver figura 4.4) donde se encuentra el archivo con el que se va a trabajar.

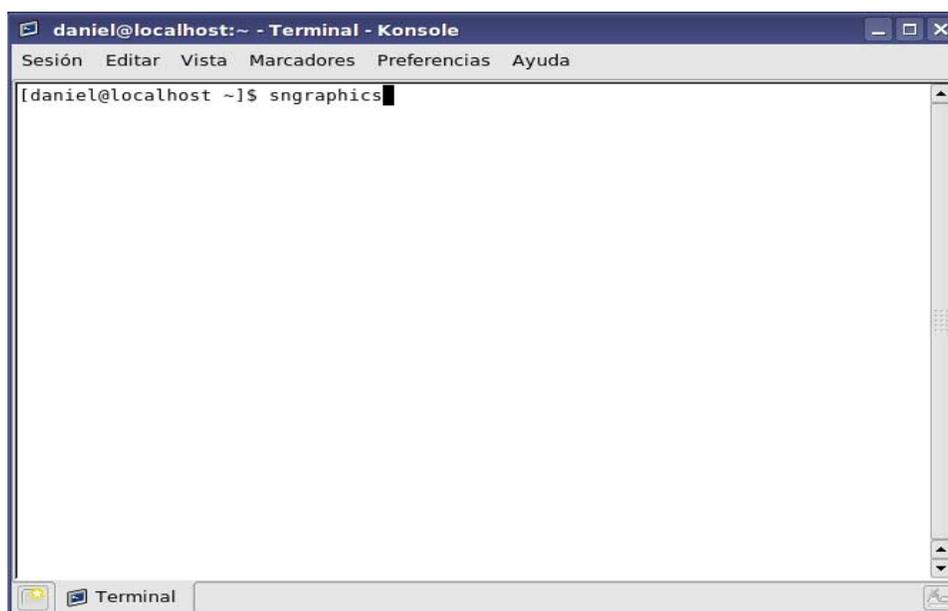


Figura 4.2: Ejecución del sistema.

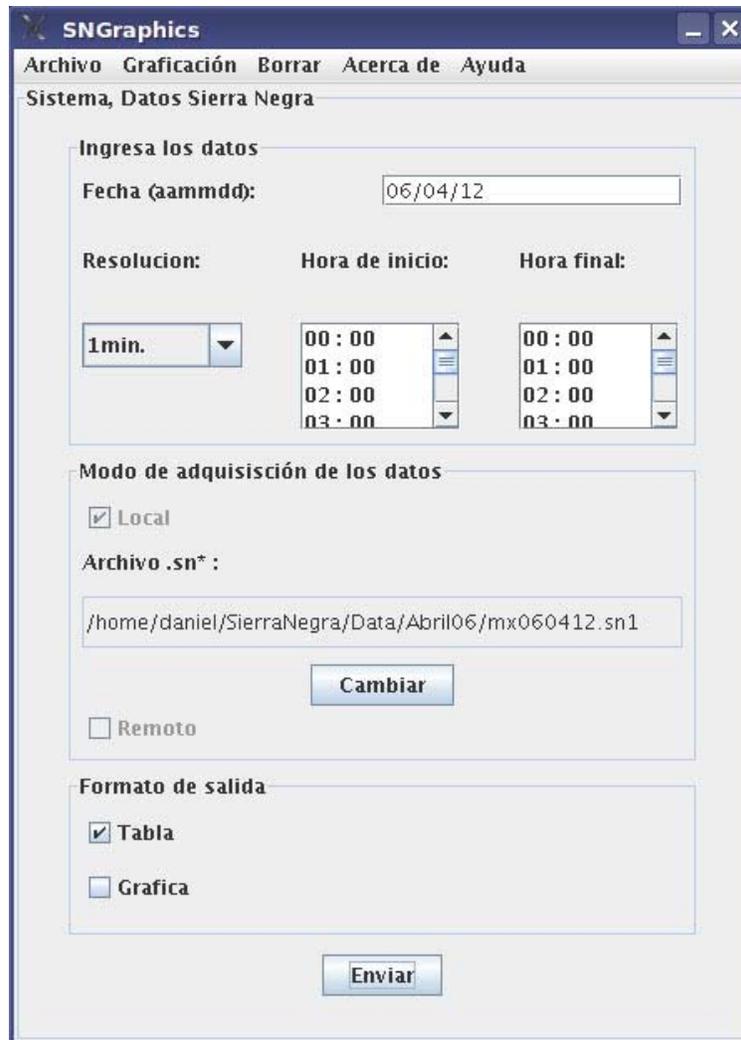


Figura 4.3: Ventana principal.

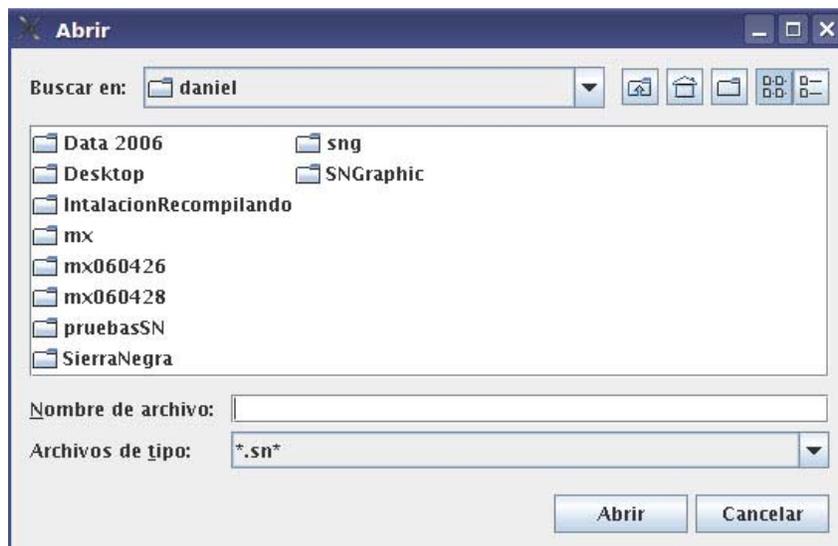


Figura 4.4: Ventana para indicar la ruta del archivo de datos crudo.

La otra opción es la forma remota, en la cual el sistema se conecta a un servidor donde se alojan los datos y extrae el archivo de la fecha indicada.

La última elección que debe hacer el usuario es el formato de salida en el cual desea visualizar los datos, y solo hay dos opciones, gráfica o tabla.

Cabe mencionar que el sistema trabajará con archivos de un solo día cuando el usuario interactúe con la ventana principal del sistema, después se explicará como generar una gráfica de dos días contiguos.

4.2.3 El menú de opciones.

El menú de opciones consta de los siguientes elementos:

1) Archivo.

- a) **Salvar 4 canales:** Esta opción nos permite salvar en un archivo de texto en alguna ubicación (directorio) deseada, los datos ocupados para generar la última gráfica o tabla de datos de un día y de 4 canales, cuya fecha se especificó en la ventana principal.
- b) **Salvar todos los canales:** Esta opción permite salvar en un archivo de texto los datos con todos los canales, en alguna ubicación (directorio) deseada cuya fecha se especificó en la ventana principal.
- c) **Salvar directorio:** esta opción permite salvar un directorio cuyo contenido son una serie de archivos de texto, y donde cada uno de ellos corresponde a los datos de la fecha especificada en la ventana principal, y de cada una de las resoluciones que se manejan en el sistema: 10 segundos, 1 minuto, 3 minutos, 5 minutos y 1 hora.
- d) **Mostrar 4 canales:** Esta opción muestra los datos de los 4 canales principales en forma tabulada, cuya fecha se especificó en la pantalla principal.
- e) **Mostrar todos los canales:** Esta opción muestra los datos de todos los canales en forma tabulada, cuya fecha se especificó en la pantalla principal.

2) Graficación.

- a) **Graficar 2 días:** Esta opción permite abrir una ventana en la cual se le piden al usuario los datos de los 2 días contiguos que desea graficar y la resolución en la que desea ver la gráfica, para poder hacer uso de esta opción se deben haber pedido con anterioridad las gráficas por día de cada una de las fechas contiguas que se desean graficar.
- b) **Graficar otros canales:** Esta opción permite al usuario ver una gráfica de de otros 4 canales cualesquiera diferentes o iguales a los 4 canales principales.
- c) **Salvar gráfica como:** Esta opción permite salvar en alguna ubicación deseada (directorio) la última gráfica generada por el sistema, en cualquiera de tres formatos, latex (.tex), postcript(.ps) y .png .

3) **Borrar.**

En esta opción el usuario puede eliminar los archivos y directorios temporales que genera el sistema, esta opción está pensada simplemente para liberar espacio en disco duro cuando el usuario lo deseé.

4) **Acerca de:** Esta opción muestra los créditos del sistema.

5) **Ayuda:** Esta opción permite visualizar una ayuda de cómo operar el sistema sngraphics.

4.2.4 Ventanas secundarias.

4.2.4.1 Ventana para salvar un archivo o gráfica.

Cuando elegimos una de las opciones de guardar, el sistema muestra una ventana, como la que se muestra a continuación, para que el usuario le indique al sistema donde quiere guardar un determinado elemento (archivos de 4 o todos los canales, directorio con archivos de todas las resoluciones y las gráficas que se van generando), esta ventana se ilustra en la figura 4.5.

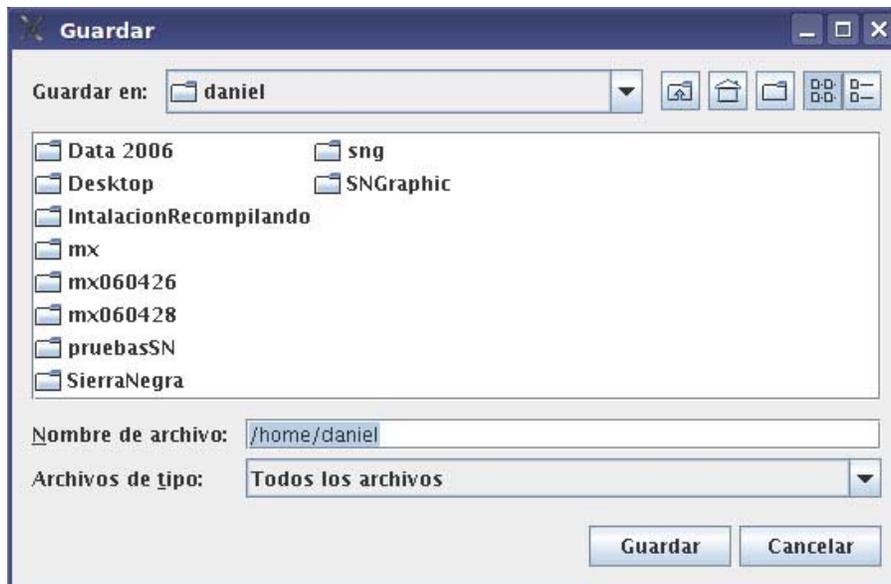


Figura 4.5: Ventana para salvar algún elemento (archivo o gráfica).

4.2.4.2 Ventana para solicitar una gráfica de dos días.

En esta ventana (ver figura 4.6) el usuario ingresa algunos datos para generar la gráfica deseada, tales datos son: dos fechas de días contiguos, la resolución, hora de inicio y hora de término de la gráfica deseada.

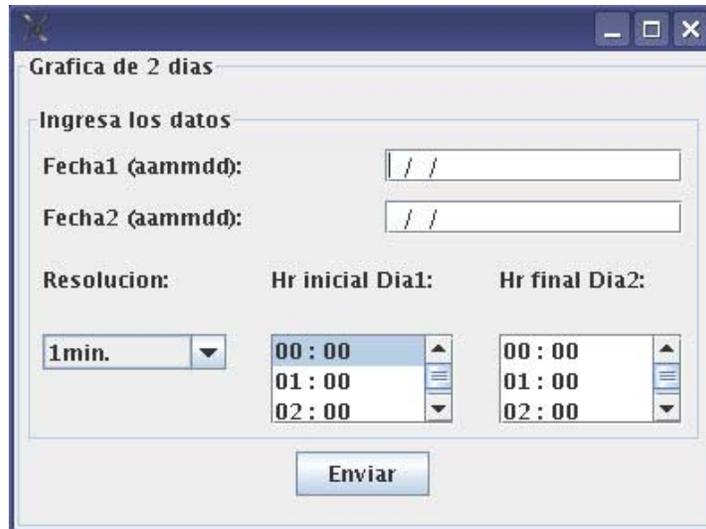


Figura 4.6: Ventana para solicitar una gráfica de dos días.

4.2.4.3 Ventana para solicitar una gráfica de otros canales.

En esta ventana el usuario elige cualesquiera 4 canales de los 45 posibles para generar una gráfica, tomando la fecha y resolución de lo capturado en la ventana principal.

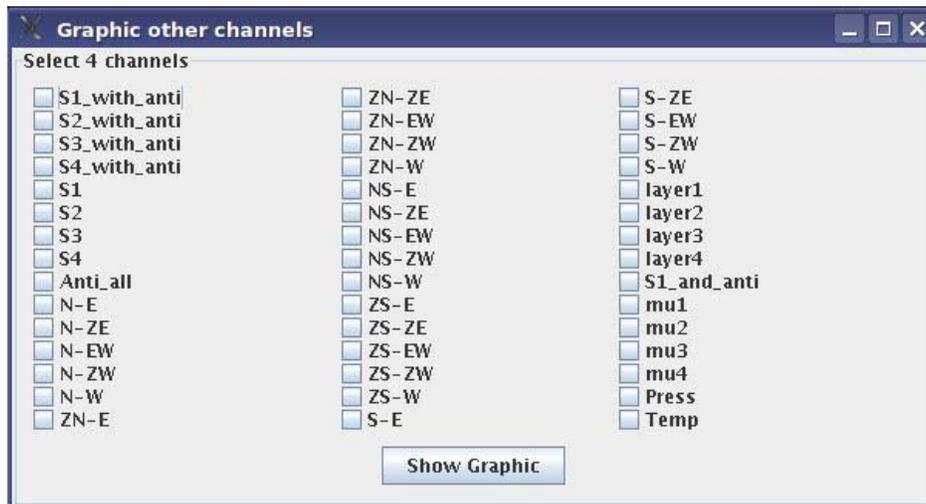


Figura 4.7: Ventana para solicitar una gráfica de otros canales.

4.3 Descripción del funcionamiento del sistema a nivel de subprogramas u objetos.

Al ejecutarse el programa sngraphics se lanza una instancia de la clase InterfazGrafica, la cual visualmente constituye la ventana principal del sistema, una vez mostrada esta ventana (ver figura 4.3), el usuario tiene que ingresar los datos que la ventana le solicita, al dar clic en el botón enviar, los datos ingresados por el usuario son capturados por el sistema, con ellos y mediante una instancia

de la clase `ProcesaDatos` con el método `verificaDatosEnTemp()` y como parámetros, los datos ingresados por el usuario, el sistema verifica si los datos solicitados ya se encuentran en el directorio de archivos temporales, si esto es afirmativo se verifica el tipo de salida de datos que eligió el usuario, si eligió tabla se invoca al método `showFromFile()` de una instancia de la clase `Dato4canales` para mostrar al usuario la tabla solicitada, o bien si el usuario eligió como forma de salida de datos una gráfica, se hace una instancia de la clase `Grafica` y mediante el método `showGraphic()` se le muestra al usuario la gráfica que pidió.

Si el método `verificaDatosEnTemp()` de la instancia de la clase `ProcesaDatos` contesta con un valor `false`, quiere decir que los datos no se encuentran en el directorio temporal, por lo cual se tienen que adquirir y procesar, es decir, se hace una instancia de la clase `ProcesamientoDeDatos` la cual mediante el método `cargaDatos()`, cuyos parámetros son los datos ingresados por el usuario en la pantalla principal, hace una instancia de la clase `ObtenerDatos` y con el método `getDatos()` el sistema se encarga de hacer una copia del archivo crudo (un archivo crudo es un archivo comprimido o no, cuya extensión es `.sn1...7..` y que el conjunto de archivos o el archivo de un día contiene los datos completos de un día entero) al directorio temporal, lo descomprime si es necesario y genera todos los archivos de datos de todas las resoluciones desde 10 segundos hasta una hora. Una vez teniendo los datos de la resolución requerida en el directorio temporal entonces se verifica el tipo de salida de datos que eligió el usuario, si eligió tabla se invoca al método `showFromFile()` de una instancia de la clase `Dato4canales` para mostrar al usuario la tabla solicitada, o bien si el usuario eligió como forma de salida de datos una gráfica, se hace una instancia de la clase `Grafica` y mediante el método `showGraphic()` se le muestra al usuario la gráfica que pidió.

Por otra parte el sistema contiene un menú de opciones, las cuales se describieron anteriormente, para su funcionamiento se requiere instancias de algunas clases.

Para cualquier opción del tipo “guardar” se ocupa el método `copiaArchivo()` de la clase `ObtenerDatos`, este método copiará del directorio temporal el archivo o la gráfica solicitados por el usuario a una ubicación que el mismo haya especificado.

La opción “Mostrar 4 canales” invoca al método `showFromFile()` de la clase `Dato4canales` de igual manera la opción “Mostrar canales” invoca al método `showFromFile()` de la clase `Dato45canales`.

La opción “Graficar 2 Días” provoca que la clase `InterfazGrafica` haga una instancia de la clase `Grafica2Dias` la cual mediante el método constructor muestra una ventana al usuario en la cual se le piden las fechas de dos días contiguos, la resolución y el intervalo de tiempo, mediante una instancia de la clase `ProcesaDatos` y con el método `verificaArchivosEntemp2D()`, se determina si existen los datos de los dos días contiguos cuya gráfica solicitó el usuario, si no es así se muestra un mensaje en el que se le dice al usuario que debe solicitar primero las gráficas por día de cada una de las fechas al final se invoca al método `showGraphic` de la clase `Grafica` para mostrar la gráfica en pantalla.

La opción "Graficar otros canales" provoca que la clase InterfazGrafica haga una instancia de la clase GraficaOtrosCanales la cual mediante el método constructor se le muestra al usuario una ventana en la cual elegirá los 4 canales cuyos datos desea visualizar en una gráfica, a partir de aquí se desencadena el mismo proceso que se lleva a cabo cuando se solicita una gráfica en la pantalla principal pero ahora la salida ya no es de los 4 canales básicos sino de los 4 canales que eligió el usuario en la ventana antes comentada.

Existen una serie de instancias de algunas clases que se realizan durante todo el proceso, los cuales no se mencionan aquí, estas instancias y sus métodos por lo general realizan algunos procesos de apoyo para el sistema, que si bien son importantes no es relevante su descripción para entender el funcionamiento general del sistema.

4.4 Descripción de las clases empleadas y sus métodos.

4.4.1 Clases del paquete IG.

InterfazGrafica.

Esta clase proporciona la ventana principal del sistema, mediante la cual el usuario interactúa con el sistema para generar las gráficas o tablas deseadas, anteriormente ya se explicaron los elementos que constituyen a esta ventana.

Métodos:

- `protected InterfazGrafica()`: Este método es el constructor de la clase, su función es inicializar las variables necesarias de la clase y determinar los elementos así como la forma en que deben ir ordenados para generar la ventana principal, tales elementos son: textfields, checkbox, list, combobox, labels, panel, menuitems, buttons etc.
- `public void menuPrincipal()`: Este método se encarga de acomodar los elementos del menú de opciones apoyándose en el método `addNewMenus`.
- `public void addNewMenus(JMenu menuArchivo, JMenu menuGrafica, JMenu menuBorrar, JMenu menuAcerca, JMenu menuAyuda)`: Este método recibe como parámetros los elementos del menú y los distribuye.
- `private String[] lanzaJFileChooser(int AbrirGuardar, int TipoExtension)`: Este método proporciona las ventanas necesarias del tipo guardar como, captura la ruta y/o nombre de los archivos a guardar y los entrega en un arreglo de cadenas.
- `public static void main(String args[])`: Dado que esta clase es la inicial y principal debe llevar el método `main()` por obligación.

Clases internas:

- `private class FiltroSN extends javax.swing.filechooser.FileFilter`: proporciona un filtro para los archivos que se puedan visualizar en la ventana generada por el método `lanzaJFileChooser()`, cuando se emplea esta para que el usuario le diga al sistema donde se encuentran los archivos de tipo `.sn1.....sn6.....`

- `private class ManejadorComboBox` implements `ItemListener`: Esta clase interna proporciona un manejador para los eventos lanzados por los elementos gráficos de tipo `Combobox`.
- `private class ManejadorCheckBox` implements `ItemListener`: Esta clase interna proporciona un manejador para los eventos lanzados por los elementos gráficos de tipo `CheckBox`.
- `private class ManejadorBotones` implements `ActionListener`: Esta clase interna proporciona un manejador para los eventos lanzados por los elementos gráficos de tipo `button`.
- `public class ManejadorEventosMenu` implements `ActionListener`: Esta clase interna proporciona un manejador para los eventos lanzados por los elementos gráficos de tipo `menuItem`.

Grafica2Dias.

Esta clase proporciona la ventana de interacción con el usuario cuando este solicita una gráfica de dos días contiguos de una resolución determinada y en un intervalo de tiempo deseado.

Métodos:

- `public String ventana()`: Este método se encarga de distribuir los elementos gráficos de la ventana por medio de la cual el usuario solicitará la gráfica de dos días contiguos.

Clases internas:

- `private class ManejadorComboBox` implements `ItemListener`: Esta clase interna proporciona un manejador para los eventos lanzados por los elementos gráficos de tipo `Combobox`.
- `public class ManejadorBotones` implements `ActionListener`: Esta clase interna proporciona un manejador para los eventos lanzados por los elementos gráficos de tipo `button`.

GraficaOtrosCanales.

Esta clase proporciona la ventana de interacción con el usuario cuando este solicita una gráfica de 4 canales cualesquiera y de una resolución determinada.

Métodos:

- `public String grafica(String strDate, String strRes)`: Este método se encarga de distribuir los elementos gráficos de la ventana por medio de la cual el usuario solicitará la gráfica de 4 canales cualesquiera.
- `boolean guardaContador(int de_donde)`: Método de apoyo para llevar la cuenta de cuantos y que canales ha elegido el usuario.
- `void disminuyeContador(int de_donde)`: Método de apoyo para llevar la cuenta de cuantos y que canales ha elegido el usuario.

4.4.2 Clases del paquete Datos.

ProcesaDatos.

Esta clase proporciona una serie de métodos que verifican la existencia de determinados archivos en el directorio temporal además de ser el primer método que empieza a trabajar directamente con los datos de los archivos crudos.

- `public boolean verificaArchivosEnTemp(String date, String res, String hini, String hfin, String modoAdq, String strSalida, int c1, int c2, int c3, int c4):` Este método recibe los datos que el usuario ingreso en la pantalla principal y además los 4 canales principales en forma numérica (1, 5, 9, 40), con estos datos el método verifica si ya existe el archivo requerido en el directorio temporal, si este es el caso se ejecutan algunos métodos de esta misma clase, de lo contrario devuelve un valor false en la parte del sistema donde fue llamado para tomar alguna decisión al respecto, en caso de que los archivos existan hace una instancia de las clases Dato4canales y Dato45canales para extraer los datos de los archivos y empezar el proceso para generar la salida adecuada (gráfica o tabla).
- `public boolean verificaArchivosEnTemp2d(String f1, String f2, String res, String hini_d1, String hfin_d2, int c1, int c2, int c3, int c4):` Este método verifica si existen los archivos concernientes a dos fechas especificadas por el usuario cuando solicita una gráfica de 2 días.
- `public void generaArchivo2dias(String rutaMasNombre2dias, String strNombreMasRuta4canales1, String strNombreMasRuta4canales2, String res, int c1, int c2, int c3, int c4):` Cuando no existen los archivos en el directorio temporal este método genera el archivo necesario auxiliandose de otros métodos.
- `public static boolean existeArchivo(String strRutaMasNombre):` este método verifica si existe el archivo cuyo nombre es pasado como argumento.

ProcesamientoDeDatos.

Esta clase brinda los métodos necesarios para generar los datos cuando estos no se encuentran en el directorio temporal.

- `public void cargaDatos(String strFecha1, String strFecha2, String strResolucion, String strHoraDelnicio, String strHoraDeTermino, String strModoAdqDatos, String strRutaDatosLocal[], String strFormatoSalida, int intCanal1, int intCanal2, int intCanal3, int intCanal4):` Este método se encarga de copiar los datos crudos desde la fuente que el usuario proporcionó (directorio local o fuente remota) y determinar si hace falta descomprimirlos o no.
- `private boolean descomprimir(String strNombreArchADescomp):` Esté método se encarga de descomprimir el archivo crudo cuando es necesario.

ObtenerDatos.*Métodos:*

Esta clase brinda los métodos para copiar los datos desde una ubicación local o remota.

- `public boolean getDatosRemotos(String strDirectorioEntemp1, String strDirectorioEntemp2, String strModoAdqDatos, String strRutaDatosLocal[])`: Obtiene los datos desde una ubicación remota.
- `public boolean getDatosLocales(String strDirectorioEntemp1, String strDirectorioEntemp2, String strModoAdqDatos, String strRutaDatosLocal[])`: Obtiene los datos crudos desde una ubicación local proporcionada por el usuario.
- `public static void copiaArchivos(String fromFileName, String toFileName) throws IOException`: Este método se encarga de copiar cualquier archivo, se requiere la ruta de origen y la destino.

GeneraArchivoNecesario.

Esta clase proporciona métodos para generar los archivos de las diferentes resoluciones a partir del archivo de datos crudo.

Métodos:

- `public String generaArchivo(String strDirectorioEntemp1, String strDirectorioEntemp2, String strResolucion, int intCanal1, int intCanal2, int intCanal3, int intCanal4)`: En base a los datos proporcionados por el usuario este método organiza la generación de los archivos de distintas resoluciones apoyándose de otros métodos de esta misma clase.
- `public void generaArchivoNecesarioF2()//se generará el archivo necesario a partir del archivo de 10 segundos formato linux`
`generaArchivo10segF2();`
`generaArchivo1min();`
`generaArchivo3min();`
`generaArchivo5min();`
`generaArchivo1hr();`
`generaArchivoParaGraficar(int4canales[0], int4canales[1], int4canales[2], int4canales[3]);`

`}`

El presente método apoyado de otros 6 métodos genera los archivos de todas las resoluciones, cuando el formato de los datos crudos corresponde al generado por el sistema de adquisición de datos en linux.

- `public void generaArchivoNecesarioF1()//se generará el archivo necesario a partir del archivo de 10 segundos formato solaris`

```

generaArchivo10segF1();
generaArchivo1min();
generaArchivo3min();
generaArchivo5min();
generaArchivo1hr();
generaArchivo1dia();
generaArchivoParaGraficar(int4canales[0], int4canales[1], int4canales[2],
int4canales[3]);

}

```

El presente método apoyado de otros 6 métodos genera los archivos de todas las resoluciones, cuando el formato de los datos crudos corresponde al generado por el sistema de adquisición de datos en solaris.

- `public void generaArchivoParaGraficar(int intCanal1, int intCanal2, int intCanal3, int intCanal4)`: Este método se encarga de extraer los cuatro canales deseados para mostrarlos en una tabla o gráfica de una resolución determinada.
- `public void salvaDatos4cEnArchivo(Dato4canales[] datosAlImprimir, String strRutaMasNombre)`: Este método es el encargado de salvar los datos extraídos para cada resolución en un archivo de texto, solo 4 canales.
- `public void salvaDatosRaizEnArchivo(Dato45canales[] datosAlImprimir, String strRutaMasNombre)`: Este método es el encargado de salvar los datos extraídos para cada resolución en un archivo de texto, 45 canales.

Dato4canales.

Esta clase brinda el manejo de los datos de cuatro canales de cualquier resolución.

Métodos.

- Métodos set.


```

public void setAnyo(String stranyo)

public void setMonth(String strmonth)

public void setDay(String strday)

public void setHour(String strhour)

public void setMin(String strmin)

public void setSeg(String strseg)

public void setCanal(int intNumCanal, int intValorCanal)

```

Estos métodos se encargan de guardar los datos de 4 canales (año, mes, día, hora, minutos, segundos, canal1, canal2, canal3 y canal4), es importante aclarar que para el almacenamiento de los canales se utiliza un tabla tipo hash.

- Métodos get.

```
public String getAnyo()

public String getMonth()

public String getDay()

public String getHour()

public String getMin()

public String getSeg()

public int getCanal(int intChannel)
```

Estos métodos se emplean para proporcionar el dato almacenado cuando este sea requerido (año, mes, día, hora, minutos, segundos, canal1, canal2, canal3 y canal4).

- `public static Dato4canales[] readFromFile(String strArchivo4Canales, String strResol, int intCanal1, int intCanal2, int intCanal3, int intCanal4)`: Este método lee los datos de cuatro canales de cualquier resolución, de un archivo y los almacena en un objeto de tipo Dato4canales.
- `public static void showFromFile(String strArchivo4Canales, String strResol,int intCanal1, int intCanal2, int intCanal3, int intCanal4)` : Este método lee los datos de 4 canales desde un archivo y los muestra en pantalla como una tabla.
- `public static void show(Dato4canales[]dato4c, String strResol,int intCanal1, int intCanal2, int intCanal3, int intCanal4)`: Este método muestra una tabla con datos de 4 canales de cualquier resolución a partir de un objeto de tipo Dato4canales.
- `public static void save(Dato4canales[] dato4c, String strDate, String strResol, String strRutaMasnombre, int intCanal1, int intCanal2, int intCanal3, int intCanal4)`: Este método se encarga de guardar los datos de 4 canales de cualquier resolución a partir de un objeto de tipo Dato4canales en un archivo de texto.
- `public void genera4CanalesFromResolucion(String strNombreMasRutaResolucion, String strNombreMasRuta4canales, String strFecha, String strResol,int intCanal1, int intCanal2, int intCanal3, int intCanal4)`: Este método genera un archivo de 4 canales de datos a partir de un archivo con 45 canales de datos.

Dato45canales.

Métodos:

- Métodos set.

```
public void setAnyo(String stranyo)

public void setMonth(String strmonth)

public void setDay(String strday)

public void setHour(String strhour)

public void setMin(String strmin)

public void setSeg(String strseg)

public void setCanal(int intNumCanal, int intValorCanal)
```

Estos métodos se encargan de guardar los datos de 45 canales (año, mes, día, hora, minutos, segundos, canal1, canal2, canal3, canal4.....canal45), es importante aclarar que para el almacenamiento de los canales se utiliza un tabla tipo hash.

- Métodos get.

```
public String getAnyo()

public String getMonth()

public String getDay()

public String getHour()

public String getMin()

public String getSeg()

public int getCanal(int intChannel)
```

Estos métodos se emplean para proporcionar el dato almacenado cuando este sea requerido (año, mes, día, hora, minutos, segundos, canal1, canal2, canal3, canal4.....canal45).

- `public static void showFromFile(String strArchivoRaiz, String strResol):` Este método lee los datos de 45 canales desde un archivo y los muestra en pantalla como una tabla.

70 Capítulo 4. El Sistema: Programación, estructura y funcionamiento

- `public static void show(Dato45canales[] datoRaiz, String strResol)`: Este método muestra en pantalla como una tabla a partir de un arreglo de objetos de tipo `Dato45canales[]`.
- `public static Dato45canales[] readFromFile(String strArchivoRaiz, String strResol)`: Este método lee los datos de 45 canales desde un archivo y los guarda en un arreglo de objetos del tipo `Dato45canales`.
- `public static void save(Dato45canales[] datosAlImprimir, String strDate, String strResol, String strPathToSave)`: Este método se encarga de guardar los datos de 4 canales de cualquier resolución a partir de un objeto de tipo `Dato45canales` en un archivo de texto.
- `public static String nombreCanal(int intCanal)`: Este método se encarga de traducir el índice de cualquier canal (1...45) en el nombre que le corresponde.

4.4.3 El paquete Graficas.

Grafica.

Esta clase proporciona los elementos para generar las gráficas solicitadas por el usuario, cabe mencionar que en tiempo de ejecución esta clase mediante uno de sus métodos genera un script de extensión `.gpl` (gnuplot) y lo ejecuta para poder visualizar la gráfica.

Métodos.

- `public void creaGrafica(String strArchivoAGraficar, String strHoraDelInicio, String strHoraDeTermino, String strDate, String strResol, String strCanal1, String strCanal2, String strCanal3, String strCanal4, boolean boPng)`: Este método se encarga de generar el script gnuplot con los parámetros adecuados para poder realizar gráficas.
- `public void ordenaCanales(int intCanal)`: Este método sirve para ordenar los valores de los canales y con ello saber cual es el menor y cual el mayor, esto se utiliza para propósitos de graficación, ya que uno de los requisitos es saber el intervalo de valores que se va a graficar.
- `public String canalEnNombre(String strCanal)`: Este método se encarga de traducir el índice de cualquier canal (1...45) en el nombre que le corresponde.
- `public void ejecutaScriptGPL()`: Este método ejecuta el escript gnuplot para poder generar la gráfica.

Clase interna.

- `public class showGraphic extends Frame` : Esta clase tiene como objetivo mostrar en pantalla la gráfica generada.

4.4.4 Utilerias.

Este paquete contiene algunas clases que sirven de apoyo para el correcto funcionamiento del sistema.

4.4.5 Parametros.

Esta clase contiene elementos que serán constantes y recurrentes durante la ejecución del sistema, como por ejemplo algunos rótulos de etiquetas, la ruta del sistema, los elementos de los combobox, etc.

EjecutaComando: Este método permite apoyarse en la ejecución de comandos nativos de Linux cuando alguno de ellos sea requerido.

4.5 Salidas del sistema.

El sistema tiene dos formatos de salida, se pueden visualizar tablas o gráficas, esto lo decide el usuario por medio de la interfaz gráfica.

La figura 4.8 muestra un ejemplo de la salida del sistema en forma de gráfica, la gráfica es de un día.

La figura 4.9 muestra una gráfica de dos días consecutivos.

La figura 4.10 muestra la salida del sistema de un fragmento en formato de tabla de 4 canales.

La figura 4.11 muestra un fragmento de una tabla de 45 canales.

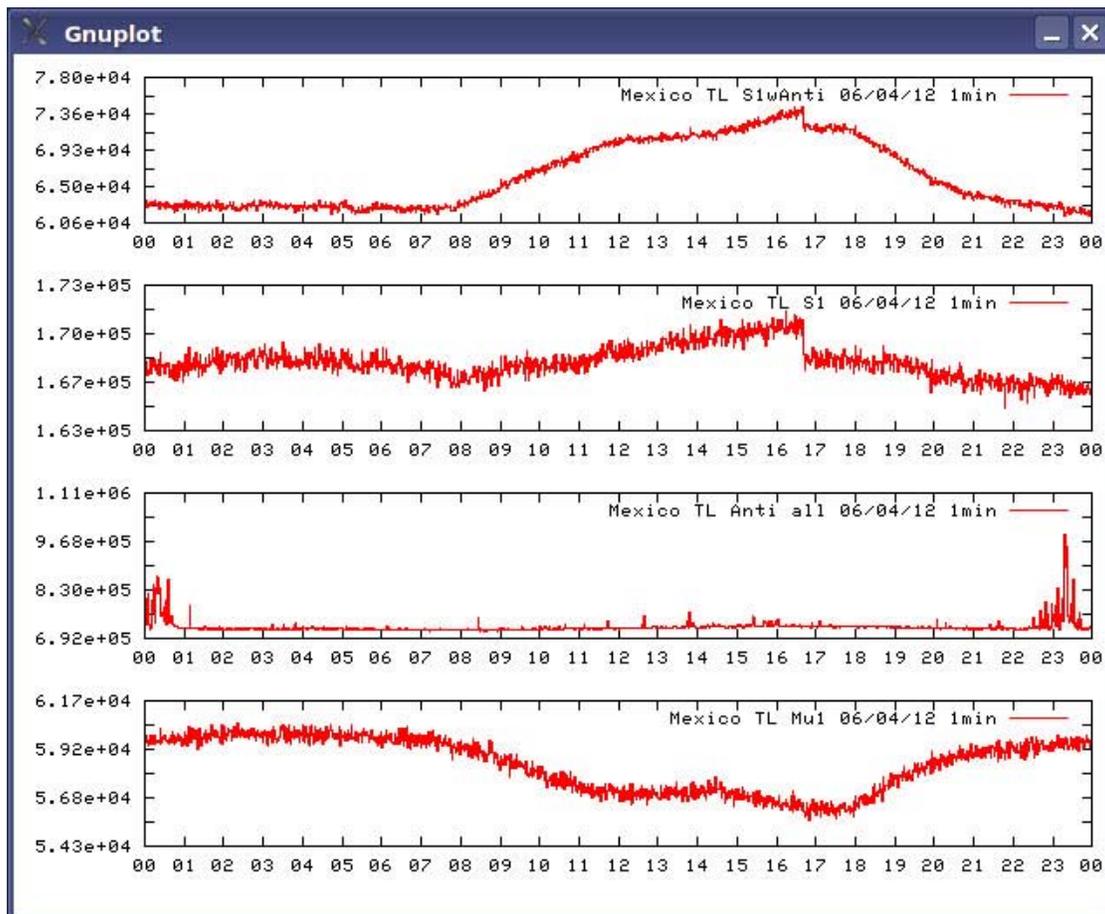


Figura 4.8: Salida del sistema en una gráfica de un día.

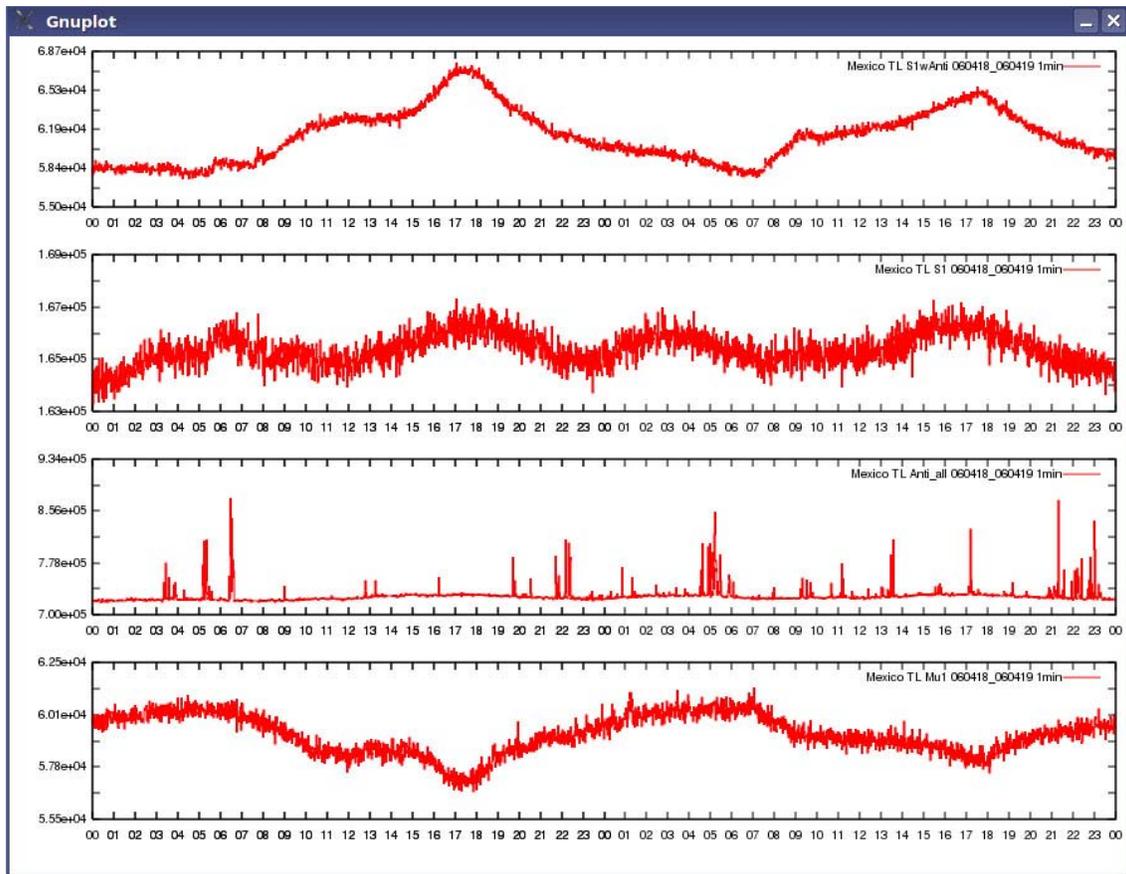


Figura 4.9: Salida del sistema en una gráfica de 2 días contiguos.

Fecha(aammdd)	Hora(hh:mm)	S1_with_Anti	S1	Anti_all	Mu1
060412	0000	62942	167482	731328	59529
060412	0001	63240	168148	730731	59959
060412	0002	63268	168048	776812	59829
060412	0003	62462	167220	806765	59610
060412	0004	62873	167520	818021	59532
060412	0005	62950	167557	746670	59419
060412	0006	62758	167436	730423	59673
060412	0007	63168	168202	718334	59921
060412	0008	62848	167687	718583	59819
060412	0009	62696	167274	719082	59474
060412	0010	63176	168209	744893	59685
060412	0011	63170	168243	801037	59905
060412	0012	62930	167739	764671	59701
060412	0013	62927	167901	846455	59530
060412	0014	62941	168026	771303	59471
060412	0015	63298	168279	765417	59811
060412	0016	62837	168294	734567	59960
060412	0017	62711	167569	835050	59609
060412	0018	62973	167910	869636	59687
060412	0019	62755	167319	820090	59657
060412	0020	63029	168280	862991	59565
060412	0021	62866	167986	864970	59663
060412	0022	62351	167212	834195	59582
060412	0023	62618	167654	785867	59507
060412	0024	63138	168744	775246	60325
060412	0025	62947	167752	755331	59720
060412	0026	62731	167871	766928	59583
060412	0027	62943	167948	762900	59806
060412	0028	62745	167756	749887	59942
060412	0029	62670	167921	740520	59852
060412	0030	62573	167287	782475	59724
060412	0031	62546	167707	796756	59879
060412	0032	62285	167461	798111	59944
060412	0033	63071	168061	732127	59656

Figura 4.10: Salida del sistema en formato de tabla de 4 canales.

Fecha(aammdd)	Hora(hh:mm)	S1_with_Anti	S2wAnti	S3wAnti	S4wAnti	S1	S2	S3
060412	0000	62942	20355	6635	3112	167482	68110	26208
060412	0001	63240	20588	6816	3103	168148	68395	26284
060412	0002	63268	20310	6675	3029	168048	68212	26052
060412	0003	62462	20081	6619	3046	167220	67834	26071
060412	0004	62873	20282	6753	3028	167520	67801	26061
060412	0005	62950	20142	6728	3109	167557	67398	25924
060412	0006	62758	20369	6689	3117	167436	67957	25993
060412	0007	63168	20478	6671	3081	168202	68471	26093
060412	0008	62848	20085	6521	2968	167687	67815	25767
060412	0009	62696	20461	6663	2980	167274	68097	26005
060412	0010	63176	20337	6735	3061	168209	68148	26047
060412	0011	63170	20258	6668	3101	168243	68094	26194
060412	0012	62930	20470	6834	3131	167739	68160	26231
060412	0013	62927	20028	6653	3135	167901	67685	25880
060412	0014	62941	20249	6627	3068	168026	67888	25832
060412	0015	63298	19986	6639	3045	168279	67929	26092
060412	0016	62837	19970	6551	3058	168294	67969	25820
060412	0017	62711	20444	6703	3050	167569	67932	25834
060412	0018	62973	20409	6812	3051	167910	67853	26009
060412	0019	62755	20118	6620	3012	167319	67576	25961
060412	0020	63029	20278	6743	3105	168280	68062	25985
060412	0021	62866	20396	6692	3008	167986	68199	26040
060412	0022	62351	20292	6678	3016	167212	68023	25855
060412	0023	62618	20178	6507	2972	167654	68039	25883
060412	0024	63138	20317	6774	3141	168744	68423	26329
060412	0025	62947	20264	6653	2997	167752	68182	26133
060412	0026	62731	20337	6599	3011	167871	68038	26147
060412	0027	62943	20210	6612	3081	167948	68305	26040
060412	0028	62745	20234	6637	2928	167756	68271	26076
060412	0029	62670	20277	6680	3018	167921	67954	25979
060412	0030	62573	20271	6742	3093	167287	67902	25922
060412	0031	62546	20232	6729	3040	167707	67980	26100

Figura 4.11: Salida del sistema en formato de tabla de 45 canales

Capítulo 5:

Uso del sistema sngraphics.



En esta sección se explica la forma de operar el sistema de manera satisfactoria, se muestran los alcances del software con una serie de ejemplos que representan todas las funciones del sistema sngraphics, así como los requerimientos de hardware y software.

5.1 Requerimientos del sistema (a nivel software y hardware).

El sistema se proyectó para su uso en plataforma Linux, específicamente en Fedora Core1, partiendo de esto se tienen los siguientes requerimientos para el adecuado funcionamiento del sistema.

Sistema Operativo: Linux, Fedora Core 1. El propósito de elegir esta plataforma y esta versión tiene su origen en que el sistema de adquisición de datos del TNS funciona bajo Linux Fedora Core 1.

El haber utilizado Linux, como plataforma, y java, como lenguaje programación, nos da la posibilidad de que el sistema sngraphics tenga el alcance de ejecutarse en cualquier distribución de Linux, siempre y cuando se instale una versión adecuada del kit de java, las distribuciones de Linux van avanzando en versiones, es decir, el kernel va superándose, adicionando nuevas cosas y mejorando otras por lo cual la versión de los paquetes a instalar también avanzan en número de versión, solo hay que cuidar que el kit de java (jdk) sea el indicado para el kernel de Linux que se esté utilizando..

Kit de Java: Se requiere (Para Linux Fedora Core 1) la versión 1.4.1 del j2sdk (j2sdk-1_4_1-linux-i586-rpm.bin). Por facilidad, se recomienda la instalación mediante el archivo .rpm. En el producto final se anexa esta versión del kit de java para que el usuario tenga todos los elementos para el buen funcionamiento del sistema sngraphics.

GNUPlot: Este software de graficación generalmente se instala acompañado del sistema operativo, si por alguna razón el usuario no lo instaló, se recomienda utilizar la versión gnuplot-3.7.3. En el producto final se anexa esta versión la cual se instala desde las fuentes¹; se anexa el software con el fin de tener todos los elementos necesarios para el adecuado funcionamiento del sistema sngraphics.

Gunzip: Se requiere también del descompresor gunzip, que viene acompañado el compresor gzip y que de igual manera que gnuplot se instala junto con el sistema operativo, al ser uno de los elementos básicos de instalación es imposible pensar que no se instale con el sistema operativo.

¹ El decir 'las fuentes' me refiero a un archivo .tar.gz, que al descomprimir se obtiene un directorio, el cual tiene elementos que permiten la compilación y posterior instalación del programa (configure y make).

Estos los requerimientos a nivel software, hablando de hardware el sistema se probó en varios equipos de tipo PC:

a) PC1: AMD-K6 a 400 MHz con 380 MB en RAM, con disco duro de 30 GB a 5200 rpm, y un espacio libre en disco de 10 GB.

b) PC2: Pentium III a 700 MHz con 512 MB en RAM, disco duro de 40 GB a 7200 rpm, y un espacio libre en disco de 25 GB.

c) PC3: Pentium IV a 1.2 GHz con 700 MB en RAM, disco duro de 120 GB a 7200 rpm, y un espacio libre en disco de 80 GB.

d) PC4: Intel Core Duo a 1.73 GHz con 2 GB en RAM, disco duro de 120 GB a 7200 rpm, y un espacio libre en disco de 90 GB.

Cabe mencionar que en la PC4 se instaló el sistema Linux Fedora Core 6, en los demás se instaló el sistema Linux Fedora Core 1.

Podemos concluir que los requisitos de hardware mínimos son los siguientes:

Computadora Personal con procesador Pentium II a 400 MHz o equivalente con mínimo 256 MB en RAM y espacio en disco duro de al menos 1 GB.

Claro está que entre más austero el equipo más se tardará el procesamiento de los datos.

5.2 Manual de Usuario.

Este apartado describe el uso del sistema, el usuario puede tener acceso a este manual mediante la opción Ayuda en html en la sección Ayuda del menú de opciones, con lo cual se desplegará la ayuda en formato html, esta ayuda se conforma de una serie de elementos descriptivos de las partes que conforman el sistema, además de una descripción general y una serie de 4 ejemplos ilustrativos acerca de cómo usar el sistema.

El contenido del manual es el siguiente:

5.2.1 Descripción general.

El sistema sngraphics tiene por objetivo el procesamiento y graficación de los datos generados por el telescopio de neutrones solares de Sierra Negra, Puebla.

Es una herramienta de software que apoya y facilita labor de investigación e interpretación de datos, que de los eventos solares, se realiza en el departamento de Ciencias Espaciales del Instituto de Geofísica de la UNAM.



Figura 5.1: Pantalla principal de menú de usuario.

Los datos que se generan en el telescopio de neutrones solares tienen varias características:

1. Actualmente son generados por un sistema de adquisición de datos en plataforma Linux, anteriormente eran generados por un sistema UNIX Solaris (datos viejos).

2. Tienen un formato de texto basado en codificación ASCII.

3. Los datos "viejos" están comprimidos, con el compresor gzip de UNIX Solaris.

Los datos "viejos" se distribuyen en 65 columnas y 8,640 filas, cada fila corresponde al muestreo de cada 10 segundos de los 65 canales de información. Se tiene un archivo por día.

El sistema de software solo trabajará con los primeros 49 canales que son los que contiene información útil.

El actual sistema de adquisición tiene 45 canales de datos útiles, adicionalmente los canales de fecha y tiempo.

El sistema cuenta con una interfaz gráfica de usuario, sencilla de usar, además de diversos módulos de procesamiento de información que permitan obtener datos de resoluciones adicionales a partir del archivo original de resolución de 10 segundos, estas otras resoluciones son:

1 minuto, 3 minutos, 5 minutos y 1 hora, el sistema es capaz de graficar datos de cualquiera de las anteriores resoluciones en el intervalo de tiempo solicitado por el usuario. El sistema tiene diversas opciones que permiten al usuario observar y guardar en ubicaciones especificadas por él, los datos y las gráficas generados por el sistema.

El sistema sngraphics es una aplicación para PC, la cual debe contar con plataforma Linux, el sistema está pensado para su uso en Linux Fedora, pero se ha comprobado su funcionamiento en Linux SUSE.

5.2.2 Pantalla Principal del Sistema.

La pantalla principal del sistema es la que se muestra en la figura 5.2:

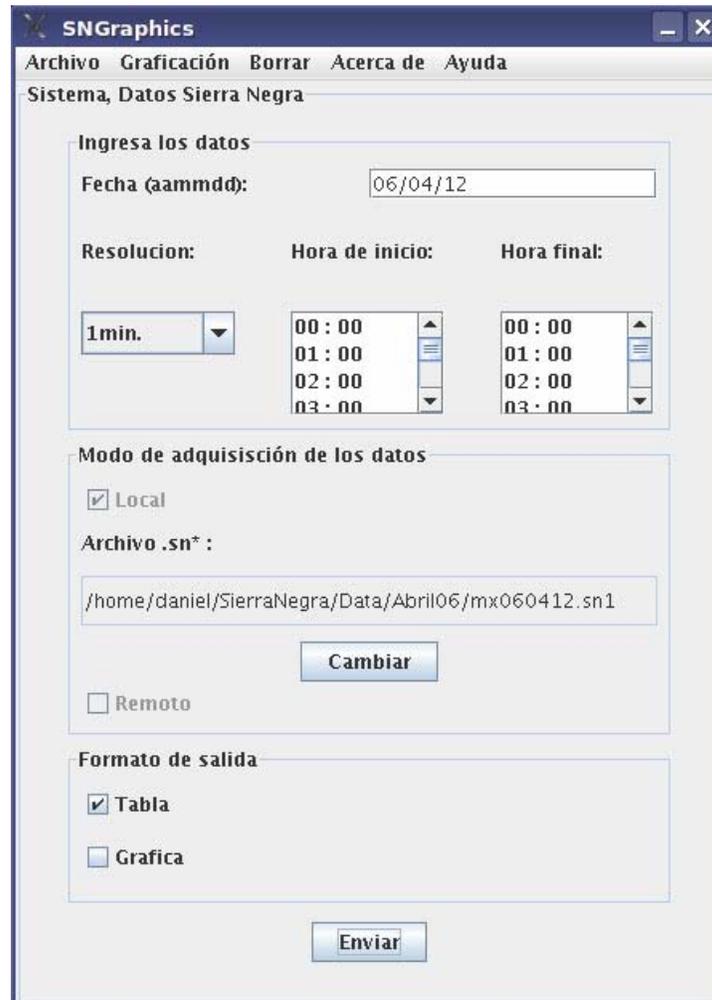


Figura 5.2: Ventana principal el sistema.

En esta pantalla podemos observar los datos que debe ingresar el usuario para solicitar una gráfica o una tabla de un día de datos o fracción de día, de una resolución determinada, los elementos se listan a continuación:

1) Campo de texto para ingresar la fecha en formato aa/mm/dd: aquí el usuario debe ingresar la fecha del día deseado, iniciando por el año en dos dígitos, seguido por el mes en dos dígitos y terminando con el día en dos dígitos.

80 Capítulo 5. Uso del sistema sngraphics

2) ComboBox de las resoluciones: El usuario debe elegir entre las diferentes resoluciones: 10 segundos, 1 minuto, 3 minutos, 5 minutos o 1 hora.

3) List de las horas: El usuario debe elegir la hora de inicio de la gráfica y la hora de término de la misma.

4) CheckBox del modo de adquisición de datos: El usuario debe elegir entre el modo local y el modo remoto para que el sistema adquiera los "Datos crudos" desde una fuente remota o local. Si el usuario elige una fuente local automáticamente se lanza una ventana de un *filechooser*, mediante la cual el usuario debe indicar al sistema de donde copiar los datos crudos, si el usuario se equivoca puede corregir este dato dando click en el botón cambiar. Si el usuario elige el modo remoto, el sistema obtendrá los datos de un servidor remoto.

5) CheckBox del formato de salida: En esta parte el usuario debe elegir la forma en que desea visualizar la salida del sistema, como gráfica o como tabla.

Todos estos elementos son proporcionados por el usuario para generar una gráfica o tabla de un día de datos o fracción de este, una vez finalizada la captura de datos se debe dar click al botón enviar para que el sistema trabaje y genere la salida deseada.

Cabe mencionar que el campo de fecha es el único en el que el usuario ingresará datos desde el teclado lo demás son elecciones entre varias opciones, esto es con la finalidad de disminuir la posibilidad de error en cuanto al ingreso de datos por parte del usuario. Si el usuario no elige ninguna de las opciones del combobox y los checkbox las opciones que se toman por default son las siguientes:

Resolución: 1 minuto.

Hora de inicio: 00:00.

Hora de término: 24:00.

Método de obtención de datos: Remoto.

Salida de datos: Gráfica.

La pantalla principal también cuenta con un menú de opciones, el cual se describirá en otra sección.

5.2.3 Gráficas.

Las gráficas son uno de los dos formatos de salida que se tienen en el presente sistema, las gráficas pueden ser de tres tipos:

1) Gráfica de los cuatro canales de día o fracción de día de una fecha y resolución determinada, esta salida se da cuando el usuario elige el tipo de formato de salida gráfica en la pantalla principal. Este tipo de gráfica se genera cuando el usuario captura los datos en la ventana principal y da click en el botón enviar.

2) Gráfica de cuatro canales cualesquiera de un día completo o fracción de día de las últimas fecha y resolución capturadas por el usuario en la pantalla principal. Este tipo de gráfica se genera cuando el usuario elige la opción "Graficar otros canales" de la parte de graficación del menú de opciones.

3) Gráfica de cuatro canales principales que abarcan 2 días de datos o fracciones de ellos y cuyas fechas deben ser contiguas, el usuario elegirá la resolución y el intervalo de tiempo que desea ver en la gráfica. Este tipo de gráfica se genera cuando el usuario elige en el menú de la opción "Graficar 2 días", en la parte de Graficación.

Independientemente del tipo de gráfica generada, el usuario puede guardar en una ubicación especificada por él la última gráfica generada.

La figura 5.3 muestra una gráfica del tipo 1, y corresponde a una gráfica de los 4 canales principales (S1 con anti, S1, Anti all y Muon1) del día 12 de Abril de 2006 y con una resolución de 1 minuto.

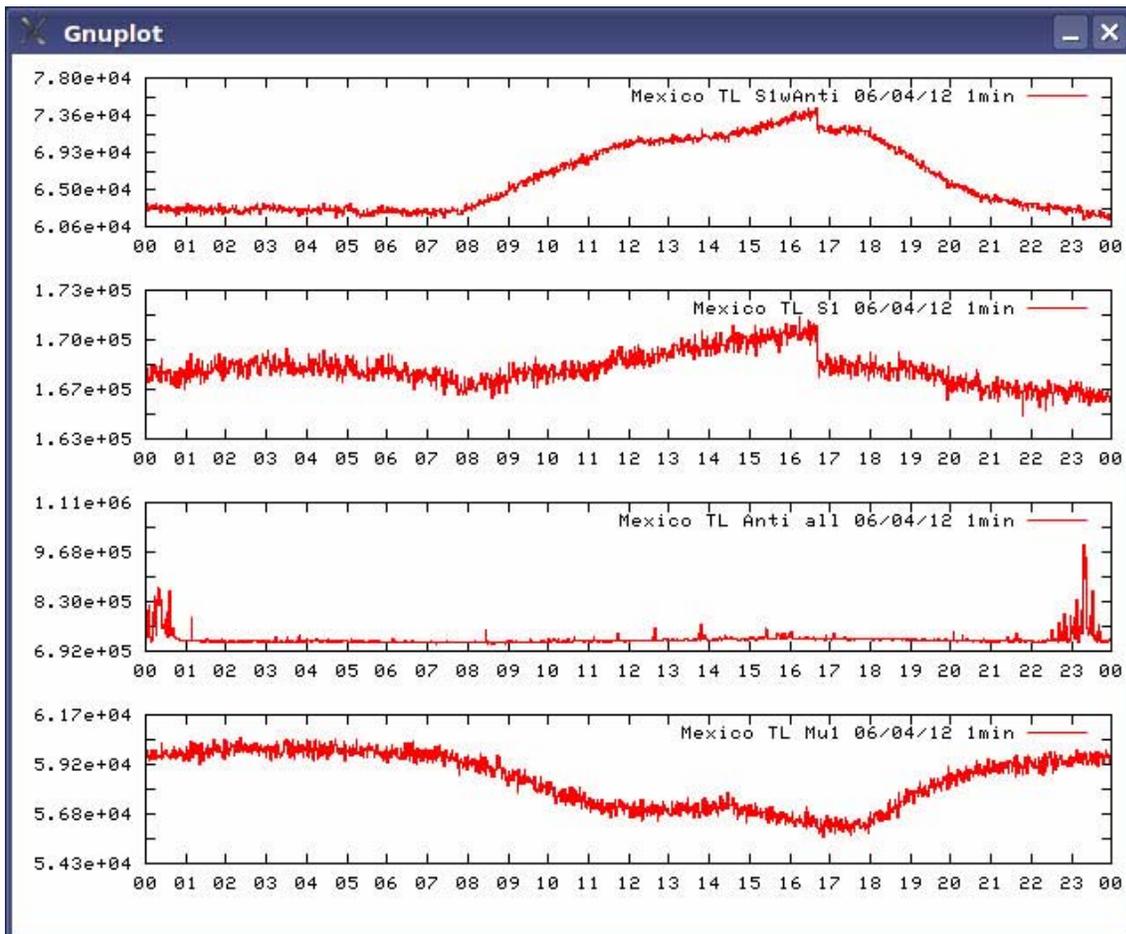


Figura 5.3: Gráfica de 4 canales principales.

5.2.4 Tablas

Las tablas son uno de los dos formatos de salida del sistema, las tablas pueden ser de dos tipos:

- 1) Tabla de los cuatro canales principales de una fecha y resolución determinada de un día completo.
- 2) Tabla de los 45 canales de una fecha y resolución determinadas de un día completo.

La figura 5.4 muestra una tabla del tipo 1.

Fecha(aammdd)	Hora(hh:mm)	S1_with_Anti	S1	Anti_all	Mu1
060412	0000	62942	167482	731328	59529
060412	0001	63240	168148	730731	59959
060412	0002	63268	168048	776812	59829
060412	0003	62462	167220	806765	59610
060412	0004	62873	167520	818021	59532
060412	0005	62950	167557	746670	59419
060412	0006	62758	167436	730423	59673
060412	0007	63168	168202	718334	59921
060412	0008	62848	167687	718583	59819
060412	0009	62696	167274	719082	59474
060412	0010	63176	168209	744893	59685
060412	0011	63170	168243	801037	59905
060412	0012	62930	167739	764671	59701
060412	0013	62927	167901	846455	59530
060412	0014	62941	168026	771303	59471
060412	0015	63298	168279	765417	59811
060412	0016	62837	168294	734567	59960
060412	0017	62711	167569	835050	59609
060412	0018	62973	167910	869636	59687
060412	0019	62755	167319	820090	59657
060412	0020	63029	168280	862991	59565
060412	0021	62866	167986	864970	59663
060412	0022	62351	167212	834195	59582
060412	0023	62618	167654	785867	59507
060412	0024	63138	168744	775246	60325
060412	0025	62947	167752	755331	59720
060412	0026	62731	167871	766928	59583
060412	0027	62943	167948	762900	59806
060412	0028	62745	167756	749887	59942
060412	0029	62670	167921	740520	59852
060412	0030	62573	167287	782475	59724
060412	0031	62546	167707	796756	59879
060412	0032	62285	167461	798111	59944
060412	0033	63071	168061	732127	59656

Figura 5.4: Tabla generada por el sistema.

5.2.5 Menú de opciones.

El menú de opciones se compone de los siguientes elementos:

5.2.5.1 Archivo.**5.2.5.1.1 Salvar 4 canales.**

Esta opción del menú de la sección de archivo, permite al usuario, salvar en una ubicación especificada por él, un archivo con los datos de los 4 canales principales correspondiente a la última fecha y resolución elegidas por el usuario en la

pantalla principal, la ventana que se abre para que el usuario indique la ubicación en donde desea guardar el archivo, es como la que se muestra en la figura 5.5:



Figura 5.5: Ventana para guardar el archivo

Una vez indicada la ruta se realiza la copia del archivo del directorio temporal a la ubicación proporcionada por el usuario.

5.2.5.1.2 Salvar canales.

Esta opción del menú de la sección de archivo, permite al usuario, salvar en una ubicación especificada por él, un archivo con los datos de los 45 canales correspondientes a la última fecha y resolución elegidas por el usuario en la pantalla principal, la ventana que se abre para que el usuario indique la ubicación en donde desea guardar el archivo, es como la que se muestra en la figura 5.6:



Figura 5.6: Ventana para guardar el archivo

Una vez indicada la ruta se realiza la copia del archivo del directorio temporal a la ubicación proporcionada por el usuario.

5.2.5.1.3 Salvar directorio.

Esta opción del menú de la sección de archivo, permite al usuario, salvar en una ubicación especificada por él, un directorio con los archivos de datos de los 45 canales de cada una de las resoluciones (10 segundos, 1 minuto, 3 minutos, 5 minutos y 1 hora) correspondientes a la última fecha y resolución elegidas por el usuario en la pantalla principal, la ventana que se abre para que el usuario indique la ubicación en donde desea guardar el directorio, es como la que se muestra en la figura 5.7:

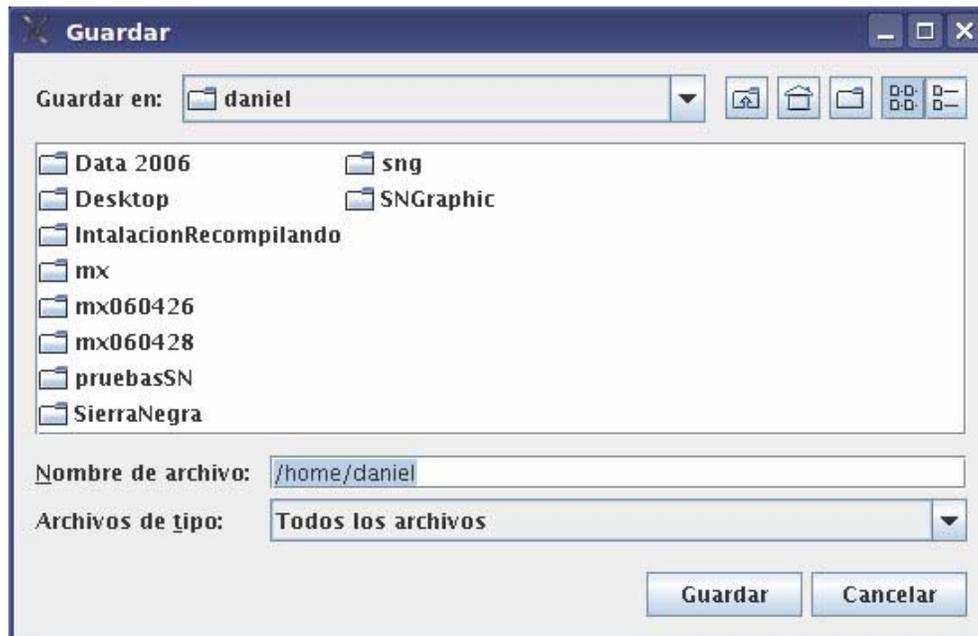


Figura 5.7: Ventana para guardar el directorio

Una vez indicada la ruta se realiza la copia del directorio con los archivos de todas las resoluciones desde el directorio temporal, a la ubicación proporcionada por el usuario.

5.2.5.1.4 Mostrar 4 canales.

Esta opción del menú permite al usuario visualizar una tabla con los datos de los 4 canales principales correspondientes a las últimas fecha y resolución capturadas por el usuario en la pantalla principal, la tabla que se muestra en este tipo de salida es como la que se muestra en la figura 5.8.

Fecha(aammdd)	Hora(hh:mm)	S1_with_Anti	S1	Anti_all	Mu1
060412	0000	62942	167482	731328	59529
060412	0001	63240	168148	730731	59959
060412	0002	63268	168048	776812	59829
060412	0003	62462	167220	806765	59610
060412	0004	62873	167520	818021	59532
060412	0005	62950	167557	746670	59419
060412	0006	62758	167436	730423	59673
060412	0007	63168	168202	718334	59921
060412	0008	62848	167687	718583	59819
060412	0009	62696	167274	719082	59474
060412	0010	63176	168209	744893	59685
060412	0011	63170	168243	801037	59905
060412	0012	62930	167739	764671	59701
060412	0013	62927	167901	846455	59530
060412	0014	62941	168026	771303	59471
060412	0015	63298	168279	765417	59811
060412	0016	62837	168294	734567	59960
060412	0017	62711	167569	835050	59609
060412	0018	62973	167910	869636	59687
060412	0019	62755	167319	820090	59657
060412	0020	63029	168280	862991	59565
060412	0021	62866	167986	864970	59663
060412	0022	62351	167212	834195	59582
060412	0023	62618	167654	785867	59507
060412	0024	63138	168744	775246	60325
060412	0025	62947	167752	755331	59720
060412	0026	62731	167871	766928	59583
060412	0027	62943	167948	762900	59806
060412	0028	62745	167756	749887	59942
060412	0029	62670	167921	740520	59852
060412	0030	62573	167287	782475	59724
060412	0031	62546	167707	796756	59879
060412	0032	62285	167461	798111	59944
060412	0033	63071	168061	732127	59656

Figura 5.8: Tabla de los 4 canales principales.

5.3.5.1.5 Mostrarm canales (todos los canales).

Esta opción del menú permite al usuario visualizar una tabla con los datos de los 45 canales correspondientes a las últimas fecha y resolución capturadas por el usuario en la pantalla principal, la tabla que se muestra en este tipo de salida es como la que se muestra a continuación.

Fecha(aammdd)	Hora(hh:mm)	S1_with_Anti	S2wAnti	S3wAnti	S4wAnti	S1	S2	S3
060412	0000	62942	20355	6635	3112	167482	68110	26208
060412	0001	63240	20588	6816	3103	168148	68395	26284
060412	0002	63268	20310	6675	3029	168048	68212	26052
060412	0003	62462	20081	6619	3046	167220	67834	26071
060412	0004	62873	20282	6753	3028	167520	67801	26061
060412	0005	62950	20142	6728	3109	167557	67398	25924
060412	0006	62758	20369	6689	3117	167436	67957	25993
060412	0007	63168	20478	6671	3081	168202	68471	26093
060412	0008	62848	20085	6521	2968	167687	67815	25767
060412	0009	62696	20461	6663	2980	167274	68097	26005
060412	0010	63176	20337	6735	3061	168209	68148	26047
060412	0011	63170	20258	6668	3101	168243	68094	26194
060412	0012	62930	20470	6834	3131	167739	68160	26231
060412	0013	62927	20028	6653	3135	167901	67685	25880
060412	0014	62941	20249	6627	3068	168026	67888	25832
060412	0015	63298	19986	6639	3045	168279	67929	26092
060412	0016	62837	19970	6551	3058	168294	67969	25820
060412	0017	62711	20444	6703	3050	167569	67932	25834
060412	0018	62973	20409	6812	3051	167910	67853	26009
060412	0019	62755	20118	6620	3012	167319	67576	25961
060412	0020	63029	20278	6743	3105	168280	68062	25985
060412	0021	62866	20396	6692	3008	167986	68199	26040
060412	0022	62351	20292	6678	3016	167212	68023	25855
060412	0023	62618	20178	6507	2972	167654	68039	25883
060412	0024	63138	20317	6774	3141	168744	68423	26329
060412	0025	62947	20264	6653	2997	167752	68182	26133
060412	0026	62731	20337	6599	3011	167871	68038	26147
060412	0027	62943	20210	6612	3081	167948	68305	26040
060412	0028	62745	20234	6637	2928	167756	68271	26076
060412	0029	62670	20277	6680	3018	167921	67954	25979
060412	0030	62573	20271	6742	3093	167287	67902	25922
060412	0031	62546	20232	6729	3040	167707	67980	26100

Figura 5.9: Tabla con todos los canales.

5.2.5.2 Graficación.

Esta opción contempla las utilidades extras respecto a la graficación de los datos

5.2.5.2.1 Graficar 2 días.

Esta opción del menú, en la sección graficación, despliega una nueva ventana en la cual el usuario tendrá que ingresar 2 fechas de días contiguos, elegir la resolución deseada así como la hora de inicio del día 1 y la hora de término en el día 2, esta ventana se muestra en la figura 5.10.

The image shows a Windows-style dialog box titled "Grafica de 2 dias". It has a standard title bar with minimize, maximize, and close buttons. The main area is titled "Ingresa los datos" and contains the following elements:

- Two text input fields for "Fecha1 (aammdd):" and "Fecha2 (aammdd):", each containing two slashes (//).
- A "Resolucion:" label followed by a dropdown menu currently showing "1min.".
- Two time selection controls: "Hr inicial Dia1:" and "Hr final Dia2:". Each consists of a list box showing "00:00", "01:00", and "02:00" with up and down arrow buttons.
- An "Enviar" button centered at the bottom of the dialog.

Figura 5.10: Ventana para solicitar una gráfica de 2 días contiguos.

Una vez ingresados estos datos y habiendo dado click en el botón enviar se muestra en pantalla la gráfica de dos días solicitada por el usuario. En la figura 5.11 se muestra una gráfica generada por la opción de graficar 2 días.

5.2.5.2.2 Graficar otros canales.

Esta opción del menú, en la sección graficación, despliega una nueva ventana en la cual el usuario tendrá que elegir 4 canales cualesquiera, los datos a graficar corresponden a la última fecha y resolución elegidas en la pantalla principal, la ventana para elegir los 4 canales se muestra en la figura 5.12, la gráfica es de un intervalo de 24 horas.

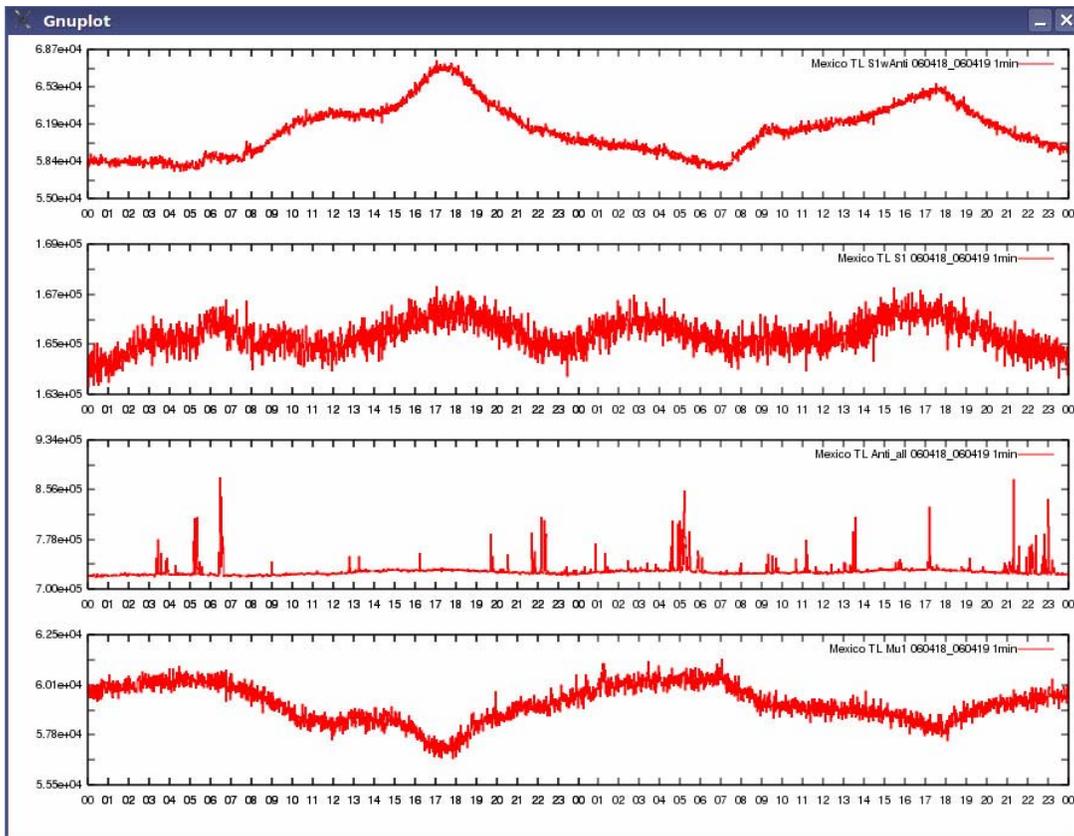


Figura 5.11: Gráfica de 2 días.

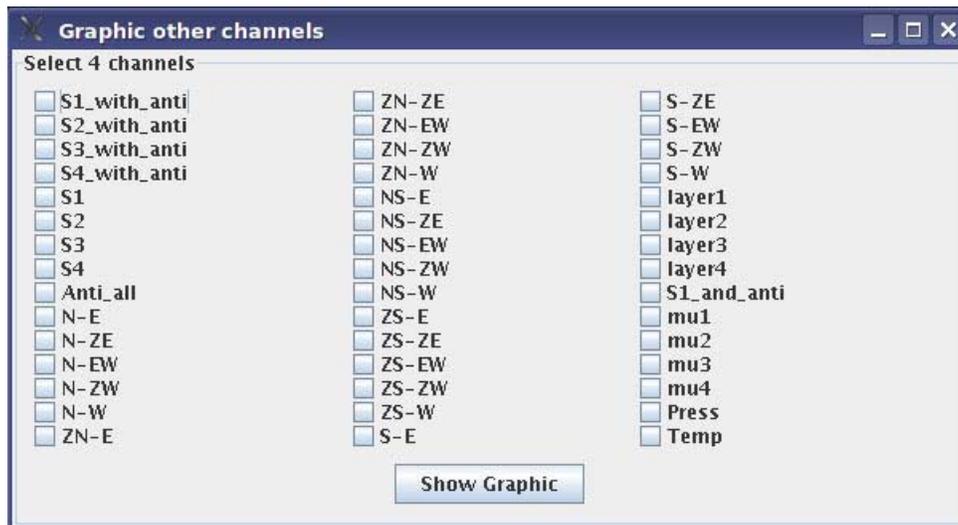


Figura 5.12: Ventana para solicitar una gráfica de otros canales.

Una vez elegidos los 4 canales y habiendo dado click en el botón enviar se muestra en pantalla la gráfica de un día de los 4 canales elegidos. En la figura 5.13 se muestra un ejemplo de este tipo de gráfica.

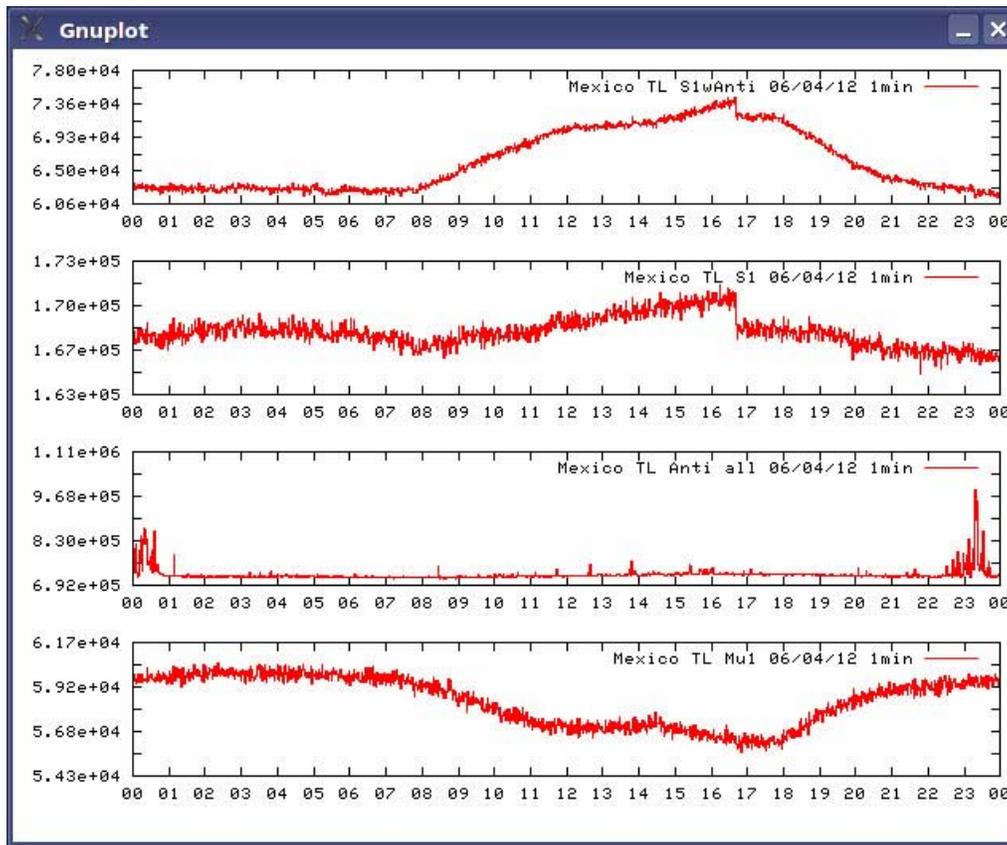


Figura 5.13: Gráfica de otros canales.

5.2.5.2.3 Salvar gráfica como: .tex, .ps ó .png

Esta opción del menú, de la sección de graficación, permite que el usuario salve en una ubicación específica la última gráfica mostrada en pantalla, se manejan tres formatos en los cuales se puede salvar la gráfica: el formato .tex (latex), .ps(postscript) y .png (formato de imagen), la ventana en la cual el usuario puede indicar la ruta donde guardar la gráfica se muestra en la figura 5.14.



Figura 5.14: Ventana para salvar una gráfica.

El resultado de indicar la ruta en la ventana mostrada, es tener la gráfica en la ubicación especificada.

5.2.5.3 Borrar.

5.2.5.3.1 Borrar temporales: El resultado de elegir esta opción es la liberación de espacio en disco mediante la eliminación de los archivos que se generaron en el directorio temporal.

5.2.5.4 Acerca de.

5.2.5.4.1 Acerca de sngraphics: El resultado de elegir esta opción es la visualización en pantalla de una ventana con los créditos del sistema.

5.2.5.5 Ayuda.

5.2.5.5.1 Ayuda en html: Al elegir esta opción se mostrará en formato html el manual de usuario.

5.3 Conclusiones.

En el proceso de desarrollo del presente trabajo puse en práctica la investigación documental, sobre todo, en el primer capítulo, debido a que el tema del Sol era totalmente nuevo para mí, aunque vital para dar a conocer al lector la justificación de la realización del trabajo escrito y del software desarrollado. Respecto al primer capítulo puedo concluir que tuve que desarrollar habilidad al momento de realizar investigación acerca de un tema que no era familiar para mí, comprendí los procesos que se llevan a cabo en el Sol, entendí la importancia del estudio de dichos procesos y sobre todo la manera en que influyen la presencia de luz y energía del Sol para el mantenimiento de la vida en nuestro planeta así como la importancia que tiene para la ciencia y para la vida cotidiana del ser humano, conocer los procesos energéticos que ocurren en nuestra estrella, las emisiones que tienen su origen en el Sol, como, todo esto afecta a nuestra vida diaria y el propósito científico del estudio de nuestra estrella.

El trabajo que presento me dio la oportunidad de conocer y adentrarme más en el tema del desarrollo del software, entender todo el proceso que se sigue para la creación de una herramienta de este tipo, que si bien teóricamente había visto algunos tópicos relativos al desarrollo de software en mis clases en la Facultad de Ingeniería, es completamente diferente ponerlos en práctica; me refiero específicamente al diseño del sistema con el UML. Tuve que realizar una serie de lecturas que me permitieran saber que era el UML, conocer la simbología que se utiliza y los diferentes diagramas que conforman el estándar del UML; una vez que logré adentrarme en el *mundo* del UML, realicé un proceso analítico para obtener los requerimientos del sistema sngraphics y para determinar cuáles de las herramientas del UML eran necesarias para el adecuado diseño del sistema. Todo esto me permitió conocer la herramienta de diseño de software del lenguaje unificado de modelado y sobre todo entender la importancia que tiene, el realizar un diseño en el cual se plasmen los requerimientos y el futuro funcionamiento del software, ya que esto facilita futuros cambios, adiciones o adecuaciones del sistema, además el estándar del UML no es restrictivo para profesionales de la computación, uno de sus propósitos principales es que cualquiera que observe los diagramas pueda entenderlos.

Partiendo de un diseño, se simplifica el proceso de programación, es un excelente intermediario entre los requerimientos expresados por el cliente y el programador del software, con ello se elimina un gran vicio que se suele tener en el desarrollo de sistemas, pensar e inmediatamente programar, cuando el proceso adecuado es: pensar, diseñar y luego programar; quizás es difícil para el profesional de la computación eliminar este vicio, pero es realmente importante poder hacerlo, que mejor que apoyarse en un estándar bastante amigable y sencillo de entender.

Tomando en cuenta que el UML se enfoca en el diseño de sistemas empleando el paradigma de orientación a objetos, para el desarrollo de este sistema, se requería un lenguaje de programación que se acoplara por completo al modelo de diseño, fue por esta razón que elegí trabajar con java, tenía conocimiento de que este lenguaje orientado a objetos tenía un sin fin de usos, pero fue hasta que inicié la programación del sistema cuando entendí realmente la forma en que trabaja java,

en su modalidad de compilador e interprete y todos los alcances que tiene, se facilita la implementación de interfaces gráficas con este lenguaje de programación. Con base en los requerimientos técnicos del sistema (manejo de cadenas, lectura de archivos de texto, implementación de una interfaz gráfica, etc) tuve que investigar las clases y métodos que me permitieran satisfacer los requerimientos técnicos mencionados. Puedo concluir que aprendí más acerca del funcionamiento del lenguaje java, conocí clases incluidas en el kit de java, sus métodos y atributos, aprendí a utilizarlas y a crear mis propias clases, en términos generales, reforcé mi habilidad para programar en lenguaje java.

Los requerimientos del sistema que obtuve después de varias entrevistas con mi tutor de tesis, incluían utilizar GNUPlot, lo que me llevó a conocer la sintaxis que emplea esta herramienta para poder elaborar las gráficas y con ello llegar a un script generalizado que me permitiera mostrar las gráficas de la manera en que mi tutor me lo solicitó. GNUPlot es una herramienta bastante amigable y sencilla de usar y sobre todo no es *pesado*, es decir, el tiempo de procesamiento es corto cuando se construye una gráfica.

Todo este aprendizaje me llevó al óptimo desarrollo del sistema sngraphics, el cual cumple con todos los requerimientos que se establecieron al inicio.

La principal fortaleza del sistema de software es que cumple con los requerimientos establecidos. De igual forma el sistema tiene algunas limitantes o inconvenientes: al utilizar java, este se vuelve *un poco lento* por la característica de java de tener una máquina virtual, es decir después de la compilación no se genera un binario que sea transparente a la plataforma (sistema operativo), sino que tiene que pasar por la JVM (Java Virtual Machine) para poder realizar la ejecución del sistema, esto lo vuelve pesado y lento.

El sistema tiene una proyección a futuro, puede complementarse con un almacenamiento de la información en una base de datos, además se puede convertir de la modalidad, aplicación para PC a una aplicación para web. De igual manera, el sistema, con algunas adecuaciones, puede ser utilizado para tabular y graficar no solamente los datos del TNS, sino cualquier tipo de datos cuyo formato sea parecido al empleado por el TNS, archivos en formato ASCII, en donde cada muestreo de cierto intervalo de tiempo corresponde a una línea de datos y la información de cada canal está separada por tabuladores.

El sistema cumple con una característica esencial de la programación orientada a objetos que es la reusabilidad, es decir, varias de las clases programadas pueden ser utilizadas en otros sistemas de software.

En términos generales aprendí sobre el Sol y los procesos energéticos que reinan en las condiciones específicas de la estrella de nuestro Sistema Planetario. También tuve la oportunidad de adentrarme más a la programación orientada a objetos, el diseño con el UML y la programación en lenguaje java, como profesionalista fue un gran aporte para mi experiencia en el rubro del desarrollo de sistemas de software.

Apéndice A.

Manual de instalación.



En este apartado se dan las instrucciones que el usuario debe seguir para la instalación de cada uno de los requisitos de software y el sistema sngraphics.

A.1 Instalación del kit de java.

Instalación del j2sdk1.4.1 para linux

0) Si usted realizó la instalación de linux completa, se instaló un compilador de java llamado gcj, por favor deshabilítelelo, de la siguiente forma.

En /usr/bin se encuentra un enlace simbólico a gcj, elimínelo.

Si no se encuentra en esta ubicación búsquelo con el comando:

```
# whereis javac
```

Lo cual desplegará la ruta donde se encuentra. Una vez ubicado el enlace simbólico, elimínelo.

Para instalar el kit de desarrollo de java j2sdk1.4.1, realice lo siguiente:
Se debe hacer todo desde una línea de comandos (shell).

1) Ingrese como root

2) Copie del CD-ROM el archivo j2sdk-1_4_1-linux-i586-rpm.bin al directorio /usr

3) Coloquese en /usr (# cd /usr) y cambie los permisos, si está en una terminal teclee:

```
# chmod +x j2sdk-1_4_1-linux-i586-rpm.bin
```

4) Ejecute el binario:

```
# ./j2sdk-1_4_0-linux-i386-rpm.bin, con ello se genera el .rpm
```

5) Tecleee:

```
# rpm -ivh j2sdk-1_4_1-fcs-linux-i586.rpm
```

Con esto ya está instalado el kit de desarrollo de java.

6) Salga del modo root.

7) Ubicándose en su directorio home, por ejemplo, el mío es: /home/daniel, desde una línea de comandos abra el archivo .bashrc, con emacs o gedit, de la siguiente forma:

```
# emacs .bashrc &
```

Agregue en la ultima línea del archivo .bashrc lo siguiente:

```
CLASSPATH=/usr/java/j2sdk1.4.1/bin  
export CLASSPATH  
export PATH=/usr/java/j2sdk1.4.1/bin:$PATH
```

esto si java quedo instalado en /usr/java

8) Salve los cambios y cierre el archivo

9) Ya está disponible el compilador y el intérprete de java

A.2 Instalación de GNUPlot.

Se debe hacer todo desde una terminal (shell)

0) Si usted eligió instalación completa al instalar linux, ya no es necesario instalar gnuplot ya que viene por defecto al instalar Linux. A veces aunque no se instale de manera completa el Linux fedora core1, también puede quedar instalado el gnuplot, para verificar si está instalado, en una línea de comandos escriba:

```
# gnuplot
```

Lo cual le deberá cambiar el prompt a:

```
#gnuplot>
```

Si sucede esto ya no necesita instalarlo, si no sucede así y aparece un mensaje de error: "command not found", se debe instalar.

Para instalarlo se deben seguir los siguientes pasos:

1) Copiar del CD-ROM el archivo gnuplot-3.7.3.tar.gz en la carpeta donde vaya a instalar gnuplot

2) Cambiar los permisos con:

```
# chmod 757 gnuplot-3.7.3.tar.gz
```

3) Descomprimirlo, así:

```
# tar -zxvf gnuplot-3.7.3.tar.gz,  
para generar la carpeta gnuplot-3.7.3
```

4) Entrar a la carpeta gnuplot-3.7.3

5) Teclear: ./configure [enter]

6) Teclear: make [enter]

7) Teclear: make install[enter]

8) Ubicarse en su directorio home, por ejemplo, el mío es: /home/daniel

9) Abrir el archivo .bashrc con emacs o gedit, agregar al final la siguiente línea:
export PATH=ruta_absoluta_donde_se_instaló_gnuplot:\$PATH
por ejemplo yo instale gnuplot en: /home/daniel/gnuplot-3.7.3, entonces, en mi caso,
ruta_absoluta_donde_se_instaló_gnuplot=/home/daniel/gnuplot-3.7.3

Con estas acciones ya tenemos instalado gnuplot.

A.3 Instalación de SNGraphics.

1) Copie desde el CD-ROM la carpeta installsg a cualquier carpeta (preferentemente /usr/local/src), deberá estar como root para realizar esto.

2) Desde una línea de comandos y estando en /usr/local/src/installsg, ejecute el programa install, de la siguiente forma:

```
#!/install
```

3) Esto instalará el programa sngraphics en /usr/local/sng

4) Respalda el archivo /etc/profile, ahora edite como root el archivo /etc/profile, antes de la línea export PATH..... Agregar:

```
PATH=$PATH:/usr/local/sng/System  
CLASSPATH=$CLASSPATH:/usr/local/sng/System
```

5) Guardar el archivo /etc/profile

6) Reiniciar la computadora.

7) Desde una terminal teclear:

```
#sngraphics
```

Y listo, se ejecutará la ventana principal del sistema.

Glosario de términos.

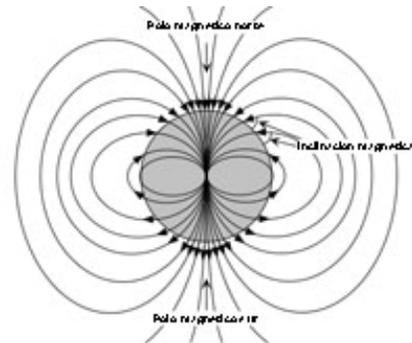
Bytes Codes: Es un código de máquina virtual que es interpretado por el intérprete Java. No es el código máquina directamente entendible por el hardware, y ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

Campo magnético: Un campo magnético es un campo de fuerza creado como consecuencia del movimiento de cargas eléctricas (flujo de la electricidad). La intensidad de un campo magnético se mide en *Gauss* (G), unidad para el sistema cgs o *Tesla* (T), unidad para el sistema MKS. El flujo decrece con la distancia a la fuente que provoca el campo.

Clase: Es una abstracción que define la funcionalidad y los elementos que constituyen a una serie de objetos.

Deuterio: Su símbolo es 2H , es un isótopo estable del hidrógeno, El núcleo del deuterio está formado por un protón y un neutrón (el hidrógeno tiene un solo protón).

Dipolo magnético: Cuando hablamos de un campo magnético, este suele definir una forma o comportamiento por medio de las líneas de campo magnético, el llamado dipolo magnético se define como “una aproximación que se hace al campo generado por un circuito cuando la distancia al circuito es mucho mayor a las dimensiones del mismo”. Su configuración se ilustra en la figura.



Espectroscopia: Es el estudio del espectro luminoso de los cuerpos, con aplicaciones en química, física y astronomía, entre otras disciplinas científicas. El análisis espectral en el cual se basa, permite detectar la absorción o emisión de radiación electromagnética de ciertas energías, y relacionar estas energías con los niveles de energía implicados en una transición cuántica.

STELAB: Solar-terrestrial Environment Laboratory, Nagoya University. ESTELAB es el acrónimo en inglés de: Laboratorio del entorno Solar-terrestre.

eV: Es una unidad de energía equivalente a la energía cinética que adquiere un electrón al ser acelerado por una diferencia de potencial en el vacío de 1 voltio. Dicho valor se obtiene experimentalmente por lo que no es una cantidad exacta. $1\text{eV} = 1.602176462 \times 10^{-19} \text{ J}$.

Fusión nuclear: Es el proceso mediante el cual dos núcleos atómicos se unen para formar uno de mayor peso atómico.

GB: Son la siglas de Giga Byte.

GNU: Es un acrónimo recursivo que significa: No es Unix (*GNU is Not Unix*). El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

GUI: Significa, interfaz gráfica de usuario (en inglés Graphical User Interface)

Ion: Un ion es una partícula que se forma cuando un átomo neutro o un grupo de átomos ganan o pierden uno o más electrones. Un átomo que pierde un electrón forma un ion de carga positiva, llamado catión; un átomo que gana un electrón forma un ion de carga negativa, llamado anión.

Java: Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

JDK: Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java.

JVM: Una Máquina virtual Java (en inglés *Java Virtual Machine*, JVM) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

Nucleon: Es una partícula componente del núcleo atómico: neutrones y protones. Los nucleones son los protones y neutrones existentes en el núcleo atómico.

Objeto: Un objeto es una cosa, generalmente extraída del vocabulario del espacio del problema o del espacio de la solución. Todo objeto tiene un nombre (se le puede identificar), un estado (generalmente hay algunos datos asociados

a él) y un comportamiento (se le pueden hacer cosas a objeto y él puede hacer cosas a otros objetos.

Programación orientada a objetos: Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

Sistema operativo: Es un conjunto de programas que controla la ejecución de programas de aplicación y actúa como una interfaz entre el usuario y el hardware de una computadora, esto es, un Sistema Operativo explota y administra los recursos de hardware de la computadora con el objeto de proporcionar un conjunto de servicios a los usuarios del sistema.

Software: Equipamiento lógico o soporte lógico, es el conjunto de los componentes (programas) necesarios para hacer posible la realización de una tarea específica en una computadora.

TNS: Telescopio de Neutrones Solares.

Toroide magnético: Es la forma a manera de anillo que toman las líneas de un campo magnético.

UML: Por sus siglas en inglés, *Unified Modeling Language*, en español Lenguaje Unificado de Modelado. Se trata de un lenguaje de modelado para el desarrollo de sistemas de software, también se puede definir como: Un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

BIBLIOGRAFÍA.

Capítulo 1.

El Sol, actividad solar y emisiones solares.

Benestad, Rasmus E. *Solar activity and earth's climate*, Praxis, Chichester, UK 2002.

Bravo, Silvia. *Encuentro con una estrella*, Fondo de Cultura Económica, México 1987 (La ciencia desde México, 38).

Fierro, Julieta, et. al. *La familia del Sol*. Fondo de Cultura Económica, México 1988 (La ciencia desde México, 62).

Mendoza, Blanca. *El Sol y la Tierra. Una relación Tormentosa*, Fondo de Cultura Económica 1993 (La ciencia desde México, 114).

Noyes, Robert W. *El Sol, nuestra estrella*, tr. del inglés por Patricia Brown de Cenos. 1ra. Edición, Editores Asociados Mexicanos, México 1985.

Priest, Eric. *Solar Magneto-hydrodynamics*. D. Reidel Pub. Co. Dordrecht 1982.

Referencias electrónicas.

Rayos cósmicos bisbiseos del Universo, disponible en:
<http://www.astrocosmo.cl/astrofis/astrofis-003.htm>

Observatorio Solar Mexicano de Gran Altura (OSOMEGA), Sierra Negra, Puebla, disponible en:
<http://www.geofisica.unam.mx/isyp/osga.html>

Capítulo 2.

Herramientas de software.

Deitel y Deitel. *Java, cómo programar*. Prentice Hall, 5ta. Edición, Prentice Hall, México, 2004.

Fowler, Martin, et al. *UML gota a gota*. tr. del inglés por: Teresa Sanz. Addison Wesley. México 1999.

Schmuller, Joseph. *Aprendiendo UML en 24 horas*. Prentice Hall, México 2004.

Stallman, Richard M. *Software libre para una sociedad libre*, tr. Del inglés por Jaron Rowan, Diego Sanz Paratcha y Laura Trinidad, 1ra. Edición, Traficantes de sueños, España 2004.

Wall, Kurt. *Programación en Linux, al descubierto*. 2da. Edición. Prentice Hall, Madrid, España, 2001.

Referencias electrónicas.

Alarcón, Raúl. *Diseño orientado a objetos con UML*. Grupo EIDOS Consultoría y Documentación Informática, S.L., Madrid, España, 2000 (Documento electrónico disponible en formato pdf).

Ferré Grau, Xavier, et al. *Desarrollo orientado a objetos con UML*, Madrid, España 2004. Documento electrónico disponible en: <http://www.clikear.com/manuales/uml>

Manzanedo del Campo, Miguel A. *Guía de Iniciación al Lenguaje JAVA*, Universidad de Burgos, 1999. Documento electrónico disponible en: <http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/Index.htm>

Sitio web de la *herramienta de graficación*: GNUPlot: <http://www.gnuplot.info/>

Capítulo 3.

El Telescopio de Neutrones Solares de Sierra Negra, Puebla y Diseño del sistema.

Martin Fowler, 1999, op. cit.

Joseph Schmuller, 2004, op. cit.

Referencias electrónicas.

Raúl Alarcón, 2000, op. cit.

Xavier Ferré Grau, 2004, op. cit.

Artículo en revista.

Valdés-Galicia, José Francisco. et al. "*An improved solar neutron telescope installed at a very high altitude in México*", Volumen 535, Número 3. pp 656-664. Elsevier B. V., Diciembre de 2004.

Capítulo 4.

El Sistema: Programación, estructura y funcionamiento.

Deitel y Deitel, 2004, op. cit.

Referencias electrónicas.

Sitio web de la *herramienta de graficación*: GNUPlot: <http://www.gnuplot.info/>

Capítulo 5.

Instalación y uso del sistema.

Deitel y Deitel, 2004, op. cit.